



Onderwijsgroep Professionele Opleidingen
Handelswetenschappen en bedrijfskunde

Cross-platform mobile apps met Xamarin

Bachelorproef aangeboden door
Samuel Debruyn
tot het behalen van de graad van
Bachelor in Toegepaste informatica

Bachelorproefbegeleider Odisee: **Serge Van Cleynenbreugel**
Bachelorproefmentor RealDolmen: **Nico Vermeir**

Academiejaar 2015 – 2016

Projectgegevens

Projectcode:	BP1516-SDRDEM
Stagebedrijf:	RealDolmen
Klant:	RealDolmen Education
Projectomschrijving:	Cross-platform mobile apps met Xamarin
Stagementor Odisee:	Serge Van Cleynenbreugel
Stagementor RealDolmen:	Nico Vermeir
Groepswerk:	neen

Woord vooraf

Dit project was niet mogelijk geweest zonder de ondersteuning die ik uit vele hoeken gekregen heb. Om die reden zou ik graag een dankwoordje richten aan enkele mensen.

Nico Vermeir had in dit project de rol van opdrachtgever. Elke vraag omtrent de opdracht werd meteen beantwoord en we konden zonder moeite afspreken om de voortgang van het project te bespreken. Hij staat bekend als iemand die ongelooflijk veel weet van ontwikkeling van apps voor Windows en softwareontwikkeling voor meerdere platformen met behulp van Xamarin. Dit blijkt zeer terecht want hij heeft me steeds uitgebreid bijgestaan wat dit betreft, waarvoor dank.

Serge Van Cleynenbreugel is docent bij de opleiding Toegepaste Informatica aan Odisee en doceert vakken rond softwareontwikkeling voor Windows en het ontwerp van gegevensbanken. Zijn vele ervaring hieromtrent en ook bij de opbouw van een bachelorproef is zeer nuttig gebleken. Ik ben dan ook zeer dankbaar voor alle tijd, de snelle respons en de zeer waardevolle feedback die ik van hem ontvangen heb.

Daarnaast zou ik ook graag alle docenten bedanken die me de afgelopen jaren kwalitatief onderwijs, interessante uitdagingen en ongelooflijk veel kennis bijgebracht hebben.

Ook Damiaan Peeters en Elgertsje Gastmans hebben veel bijgedragen aan mijn opleiding. Wat begon als een vakantiejob werd al gauw een rijke ervaring waarin ik groeide als softwareontwikkelaar en als persoon.

Tot slot wens ik ook mijn ouders en mijn vriendin te bedanken voor alle moeite die ze in mijn opleiding gestoken hebben in de vorm van ondersteuning of het nalezen en feedback geven op de vele opdrachten. Bedankt!

Inhoudsopgave

Projectgegevens	2
Woord vooraf	3
1 Inleiding	7
2 Voorstelling van de opdrachtgever	8
2.1 RealDolmen	8
2.2 RealDolmen Education	8
2.2.1 <i>Instructor-Led Training</i>	8
2.2.2 <i>E-Learning Projects</i>	9
2.2.3 <i>Technical Writing</i>	9
3 Beschrijving van de opdracht	10
4 Aanpak	11
4.1 Projectmatige aanpak	11
4.2 Fase 1: projectdefinitie	13
4.3 Fase 2: onderzoek naar oplossingsmethodes	15
4.4 Fase 3: analyses van mogelijke oplossingen	16
4.5 Fase 4: voorbereiding van implementatie	17
4.5.1 <i>Technische analyse</i>	17
4.5.2 <i>Prototype</i>	17
4.5.3 <i>Evenementen</i>	17
4.5.4 <i>Laatste voorbereiding</i>	18
5 Besluit	19
6 Zelfreflectie	20
7 Figurenlijst	21
8 Bibliografie	22
9 Bijlagen	25
9.1 Contacthistoriek	26
9.2 Tijdsbesteding	30
9.3 Presentaties	33
9.3.1 <i>Presentatie 1: pitch</i>	33
9.3.2 <i>Presentatie 2: deliverables</i>	35
9.3.3 <i>Presentatie 3: slotpresentatie</i>	37
9.4 Verslagen events	41
9.4.1 <i>Verslag AZUG: 3 sessions and a BBQ</i>	41
9.4.2 <i>Verslag AZUG: AzureCon</i>	43

9.4.3	<i>Verslag MADN: Securing and consuming a Web API with OAuth</i>	45
9.5	Project Initiatie Document	47
9.5.1	<i>Projectfiche</i>	47
9.5.2	<i>Updatehistoriek</i>	47
9.5.3	<i>Distributielijst</i>	47
9.5.4	<i>Projectdefinitie</i>	48
9.5.5	<i>Aanpak</i>	54
9.5.6	<i>Beheersaspecten</i>	56
9.6	Requirements analyse.....	57
9.6.1	<i>Functionele requirements</i>	57
9.6.2	<i>Niet-functionele requirements</i>	57
9.7	Lijst deliverables.....	58
9.7.1	<i>Bachelorproef</i>	58
9.7.2	<i>Stage</i>	59
9.8	Project planning	60
9.9	Use cases	61
9.9.1	<i>Beschrijvingen</i>	61
9.9.2	<i>Diagram</i>	65
9.10	Technische analyse.....	66
9.10.1	<i>Architectuur</i>	66
9.10.2	<i>Afhankelijkheden</i>	67
9.10.3	<i>EducationAPI</i>	69
9.11	Vergelijking cross-platform development environments	72
9.11.1	<i>Features per ontwikkelplatform</i>	72
9.11.2	<i>Core features per ontwikkelplatform</i>	73
9.12	Vergelijking native en web apps	74
9.12.1	<i>Inleiding</i>	74
9.12.2	<i>Wat is een native app?</i>	74
9.12.3	<i>Wat is een web of hybride app?</i>	76
9.12.4	<i>Vergelijking</i>	77
9.12.5	<i>Gulden middenweg</i>	78
9.12.6	<i>Conclusie</i>	78
9.13	Vergelijking Xamarin & Xamarin.Forms	79
9.13.1	<i>Introductie tot Xamarin</i>	79
9.13.2	<i>Ontwikkeling van mobiele applicaties</i>	81
9.13.3	<i>Onderscheid tussen de Xamarin-producten</i>	82
9.13.4	<i>Keuze maken tussen Xamarin (Xamarin.Android en Xamarin.iOS) en Xamarin.Forms</i>	86

9.13.5	<i>Conclusie</i>	87
9.14	API documentatie.....	88
9.15	Schermen van wireframes.....	89
9.16	Demo apps.....	90

1 Inleiding

Deze bachelorproef omschrijft hoe een probleem van een bedrijf, RealDolmen, omgezet wordt in een concreet project dat hier een oplossing aan biedt. Het project bestaat enerzijds uit deze bachelorproef en anderzijds uit een stage. Dit eerste deel bevat voorbereidend werk en onderzoek naar een goede oplossing terwijl de stage bestaat uit de implementatie van de gekozen oplossing.

Het eerste deel van dit project verliep in vier fases. Na de projectdefinitie kon er oplossingsgericht gewerkt worden. De gevonden oplossingen werden vervolgens onderling vergeleken. Tot slot moest de gekozen oplossing voorbereid worden zodat de implementatie vlot van start kan gaan.

In dit project is RealDolmen Education, de opleidingstak van softwarebedrijf RealDolmen, op zoek naar een mobiele app voor smartphones en tablets die op meerdere mobiele systemen zoals Android, iOS en Windows Phone kan draaien. Alhoewel de vraag eenvoudig lijkt, ligt een geschikte oplossing niet meteen voor de hand.

In principe is het namelijk onmogelijk om één applicatie te ontwikkelen die op meerdere besturingssystemen uitgevoerd kan worden. De vraag naar een oplossing hiervoor is groeiende waardoor enkele bedrijven er toch in geslaagd zijn om dit te realiseren. Vandaag bestaan er commerciële oplossingen die een mouw aan deze beperking passen en ze hebben elk unieke karakteristieken. RealDolmen suggereerde zelf een applicatie met een commercieel product genaamd Xamarin als oplossing en dit is ook een van de paden die gedurende het project bewandeld werden.

Om een goede oplossing aan te kunnen bieden, moet er gewerkt worden met degelijke bouwstenen. Die moeten elk kwalitatief en stabiel zijn. Dat is dan ook de reden waarom in deze bachelorproef de nadruk ligt op onderzoek en analyse, zo diepgaand als nodig. Het onderwerp leidt zich ertoe om de softwarematige architectuur te ontleden van de mobiele applicaties die vandaag gangbaar zijn en hoe daarvan gebruik gemaakt kan worden om apps voor meerdere platformen te ontwikkelen. Van termen als *bytecode of compilers* zijn de analyses dan ook niet schuw. Waar deze voor staan en hoe ze hun rol spelen in dit verhaal wordt volledig uit de doeken gedaan.

Daarnaast wordt ook het proces en de aanpak om al deze producten te realiseren beschreven. Zonder degelijke planning en regelmatige afstemming met de klant loopt een dergelijk project namelijk snel in de soep.

De stadia die het project doorlopen heeft, leverden overigens zelf ook interessante vraagstellingen op. Zo kan de vraag opgeworpen worden welke types apps bestaan en hoe ze onderling verschillen en opgebouwd zijn. Daarnaast komen er mogelijke oplossingen aan bod die elk voor- en nadelen hebben. Onder andere op deze vragen wordt dan ook dieper ingegaan met uitgebreide analyses die de keuze voor de juiste oplossing voor de vraag van de opdrachtgever begeleiden en onderbouwen.

2 Voorstelling van de opdrachtgever

2.1 RealDolmen

RealDolmen is een Belgisch bedrijf gespecialiseerd in de integratie van bestaande en eigen ICT-producten. Het bedrijf is ontstaan in 2008 na de fusie van Real Software en Dolmen en stelt vandaag meer dan 1500 mensen te werk. Het heeft afdelingen die elk gespecialiseerd zijn in een specifiek technisch domein, zoals bijvoorbeeld .NET software, Java software, netwerken en systemen, business intelligence...



Figuur 1: logo van RealDolmen

De hoofdzetel van het bedrijf is gevestigd in Huizingen. Daarnaast heeft het satellietkantoren in Kontich, De Pinte, Diegem, Harelbeke, Lummen, Namen en Bergen. Ook in Luxemburg is er nog een kantoor van Real Solutions.

Dankzij de goed verspreide kantoren is RealDolmen heel actief in lokale markten. Hier tracht het de referentie te zijn voor geïntegreerde en volledige ICT-oplossingen. Bij RealDolmen staat het eindproduct centraal, niet de technologie erachter. Zo kunnen de klanten zich concentreren op hun eigen kerntaken en wordt ICT een middel om die efficiënter te laten verlopen.

In dit project heeft RealDolmen de rol van opdrachtgever. De stage wordt door de .NET-afdeling van het bedrijf begeleid, maar de eigenlijke klant is een ander intern departement: RealDolmen Education.

2.2 RealDolmen Education

Een van de afdelingen binnen RealDolmen is het Education departement. Deze sectie verzorgt opleidingen en vorming voor zowel eigen personeel als voor externen. Andere bedrijven komen dus regelmatig aankloppen om hun medewerkers door RealDolmen Education te laten opleiden.

Wat is goede software waard indien uw eindgebruikers er niet of niet efficiënt mee kunnen werken? Hoe goed zal uw eigen infrastructuur voldoen aan de eisen van uw business indien uw systeembeheerders niet mee evolueren met de laatste trends? Hoe goed zal uw eigen software voldoen aan de eisen van uw business indien uw ontwikkelaars niet mee evolueren met de laatste trends?

RealDolmen Education

Het departement biedt verschillende diensten aan, maar de drie kerntaken zijn Instructor-Led Training (ILT), e-Learning Projects en Technical Writing. Vanzelfsprekend passen alle diensten en vormen van opleidingen binnen het technische kader waarin RealDolmen actief is op de markt.

2.2.1 Instructor-Led Training

De klassieke opleidingen met een lesgever en een groep studenten vormen de belangrijkste troef van RealDolmen Education.

2.2.2 E-Learning Projects

Naast ILT ontwikkelt RealDolmen ook digitale leeromgevingen waarin studenten zelfstandig materie kunnen aanleren, begeleid door software.

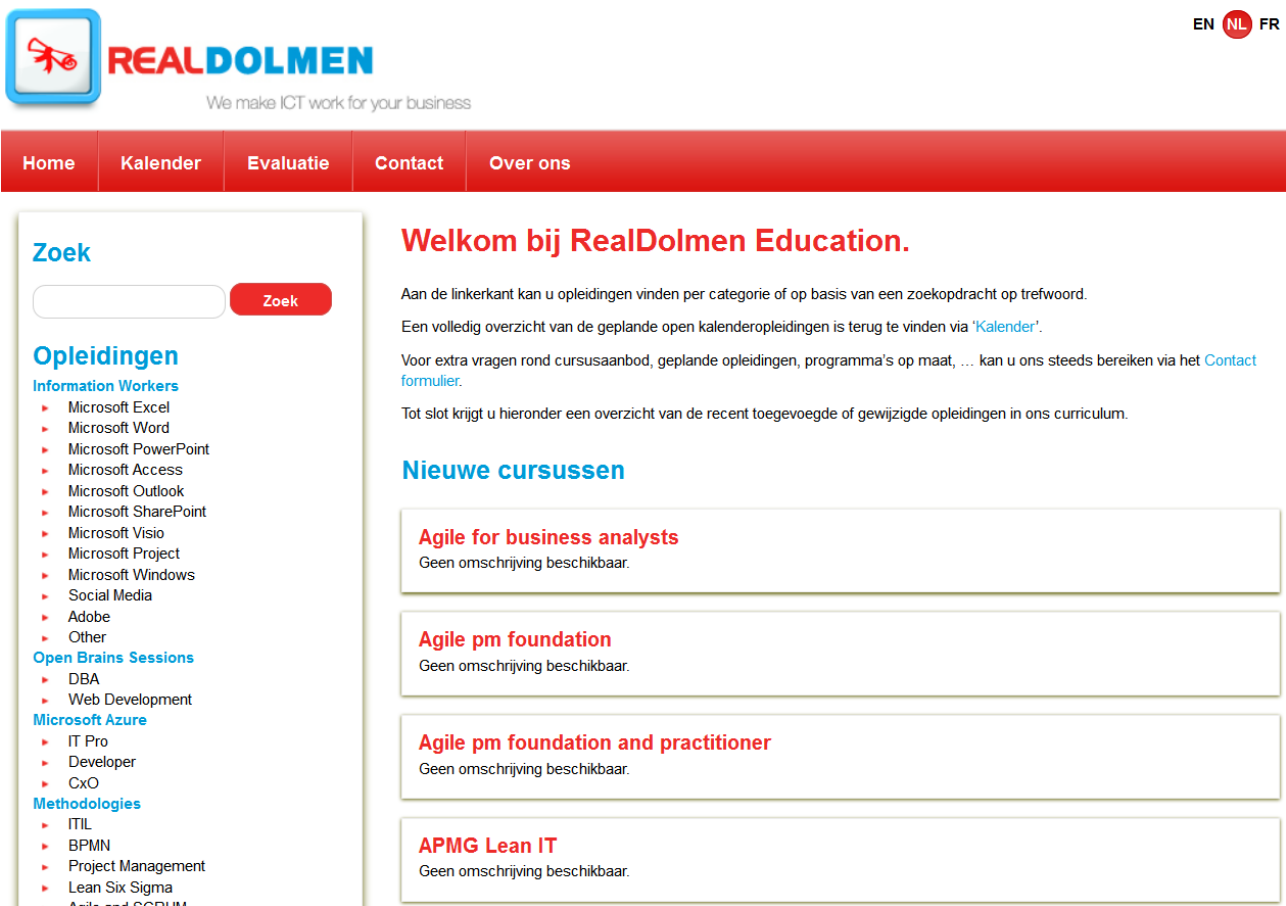
2.2.3 Technical Writing

Vaak ontbreekt het software aan goede documentatie. Hier probeert RealDolmen Education een oplossing voor te bieden. Ze kunnen zowel technische handleidingen als educatief materiaal om met software aan de slag te gaan opstellen. Zowel voor bestaande als voor software uit eigen huis.

3 Beschrijving van de opdracht

RealDolmen bundelt jaarlijks een tiental interessante stageopdrachten waarbij de klant telkens een departement van RealDolmen is, maar niet noodzakelijk hetzelfde departement als hetwelk dat de opdracht begeleidt. Zo vraagt het Education departement naar een mobiele applicatie voor meerdere platformen.

RealDolmen Education biedt ILT-opleidingen aan via de website education.realdolmen.com. Deze website laat bezoekers toe om te zoeken in het aanbod aan opleidingen, zowel via een zoekfunctie als door gebruikers toe te laten opleidingen gecategoriseerd weer te geven. De website bevat ook nog andere functies zoals een contactformulier en een mogelijkheid om feedback op een opleidingssessie te geven.



Figuur 2: de website van RealDolmen Education

Momenteel is de website niet geschikt voor gebruik op mobiele toestellen zoals smartphones en tablets. Navigeren door de categorieën verloopt moeizaam en de gebruikerservaring van de website is op deze toestellen niet vergelijkbaar met die op een desktopcomputer of laptop.

Het departement wil daarom een mobiele toepassing die soelaas brengt aan deze tekortkomingen. Het zorgt ook meteen voor een duwtje in de rug voor de naamsbekendheid van RealDolmen in applicatiewinkels op mobiele platformen. Om het potentiële aantal gebruikers van de toepassing te maximaliseren, wil RealDolmen Education de applicatie tegelijk op de populairste mobiele platformen lanceren.

De vraag van de klant wordt verder toegelicht in bijlage 9.5 PROJECT INITIATIE DOCUMENT (9.5.4 PROJECTDEFINITIE) en bijlage 9.6 REQUIREMENTS ANALYSE.

4 Aanpak

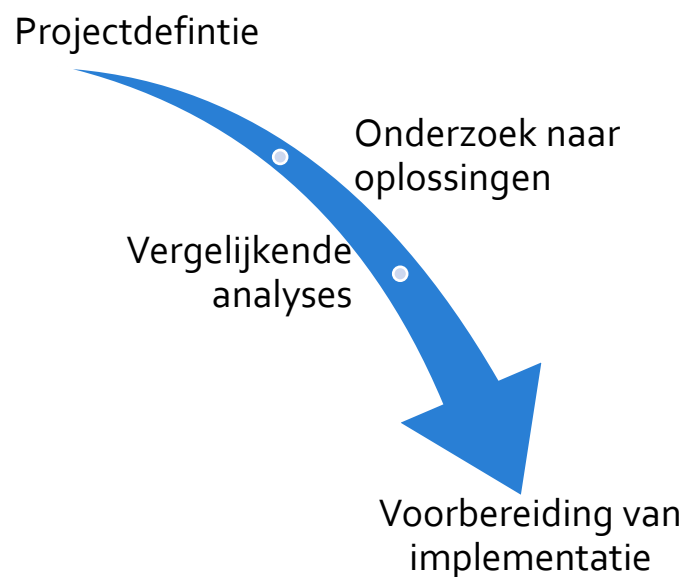
In het eerste deel, de bachelorproef, wordt een goede aanpak voor dit probleem uitgetekend. In het tweede deel, de stage, wordt de gevonden oplossing geïmplementeerd en getest. RealDolmen heeft zelf een oplossing met Xamarin¹ voorgesteld dat verder in dit hoofdstuk wordt uitgediept.

4.1 Projectmatige aanpak

Om georganiseerd en gestructureerd te werk te gaan, werd een historiek bijgehouden van de tijdsbesteding via de tool Toggl.com. Bijlage 9.2, TIJDSBESTEDING, is hier het resultaat van.

Een overzicht van de communicatie werd bijgehouden in een Excel-bestand en is terug te vinden als bijlage 9.1 CONTACTHISTORIEK.

De analyse bestaat uit vier fases of stadia. In de eerste fase werd de klant en de vraag van de klant onderzocht. Nadat de projectdefinitie duidelijk was, werden de mogelijke oplossingen onderzocht en onderling vergeleken. Vervolgens werden prototypes ontwikkeld op basis van de twee oplossingen die het beste uit de vergelijking kwamen. Ten slotte werd in een laatste stadium de implementatie van de oplossing die de klant gekozen had, voorbereid. Deze fases zijn geen strikte opsplitsingen, maar lopen vlot in elkaar over.



Figuur 3: fases van dit project

Tijdens elke fase – en vooral de eerste – werd er regelmatig met de opdrachtgever gecommuniceerd om de vraag van de klant zo duidelijk mogelijk weer te geven en de oplossing zo precies mogelijk op deze vraag te kunnen afstemmen.

Bij de voltooiing van het eerste stadium, de projectdefinitie, werd de omvang van het project duidelijker. Daarom werden alle taken, nodige analyses en op te stellen documenten weergegeven in een coherente lijst van deliverables die terug te vinden is als bijlage 9.7 LIJST DELIVERABLES.

Ten slotte kon de lijst van deliverables gebruikt worden om de planning op te stellen. Bijlage 9.8 PROJECT PLANNING geeft schematisch weer binnen welk stadium elke deliverable kan gekaderd worden en tegen wanneer deze voltooid zou moeten zijn. Zo valt ook meteen af te leiden wanneer een bepaald stadium en alle

¹ Zie 4.4 FASE 3: ANALYSES VAN MOGELIJKE OPLOSSINGEN en 9.13 VERGELIJKING XAMARIN & XAMARIN.FORMS

deelproducten volledig voltooid zijn. De meeste taken zijn onderzoeken, analyses of de uitwerking van een prototype. Hierdoor was het moeilijk inschatten welke taken het meeste werk kostten. Daarom werd voor elke taak ongeveer dezelfde hoeveelheid tijd voorzien.

Achteraf bleek dat deze aanpak goed gewerkt heeft. Alle taken waren af op de voorziene data. Na feedback werden sommige deliverables wel nog aangepast om ze beter af te stemmen op de vraag van de klant. Dit blijkt ook uit de historiek van de tijdsbesteding en de contacthistoriek. Aan respectievelijk het aantal gespendeerde uren en verzonden e-mails is af te leiden dat de hoeveelheid werk daalde in december aangezien het meeste in oktober en november al voltooid was. Dit is visueel weergegeven in **Fout! Verwijzingsbron niet gevonden.** en **Fout! Verwijzingsbron niet gevonden..** Vanaf eind december werd er niet meer aan de deliverables van het project gewerkt, enkel aan de integratietekst.

4.2 Fase 1: projectdefinitie

In de eerste fase werden documenten opgesteld om de vraag van de klant, RealDolmen Education, zo nauwkeurig mogelijk in kaart te brengen. Om dit te realiseren werden er drie gesprekken gevoerd met de opdrachtgever, RealDolmen, en lagen alle voorlopige versies van de documenten ter goedkeuring voor.

Vooreerst was het belangrijk om de huidige situatie van de klant te leren kennen. Aangezien de gevraagde mobiele toepassing gedeeltelijk ter vervanging en ter aanvulling van de huidige website moet dienen, werden alle mogelijkheden van de huidige website bestudeerd. Zo kunnen bezoekers op de website zich inschrijven voor een opleiding, maar moeten ze daarbij heel wat persoonlijke gegevens invullen. In de mobiele equivalent is hiervoor een authenticatiemodule nodig die klanten kan herkennen op basis van hun gebruikersnaam en wachtwoord.

Tijdens een vorig project werd er al een API² ontwikkeld waarvan de mobiele applicatie gebruik moet maken voor de overdracht van gegevens (opleidingen, data, informatie over opleidingen, contactfunctie en dergelijke). Alle functies die voor de mobiele app voorzien (moeten) zijn, moeten dus beschikbaar zijn in deze API. Om dit na te gaan werd de broncode van de API geanalyseerd en gedocumenteerd. Het product hiervan is terug te vinden in bijlage 9.14 API DOCUMENTATIE. Deze documentatie kan ook ingedeeld worden onder fase 4 (voorbereiding van implementatie) aangezien de implementatie aan de hand hiervan vlotter kan verlopen.

De gevraagde vereisten voor de mobiele toepassing werden gedocumenteerd in bijlage 9.6 REQUIREMENTS ANALYSE. Hierin is duidelijk weergegeven wat de mogelijkheden moeten zijn van de applicatie en hoe deze moeten werken. Enkele belangrijke vereisten zijn de mobiele platformen (Google Android, Apple iOS, Microsoft Windows Phone) en hun respectievelijke versies waarvoor er een mobiele applicatie moet zijn. Ook de manier waarop de mobiele app verschilt per mobiel platform werd nader bepaald. Zo vraagt RealDolmen om de gebruikerservaring die eigen is aan een mobiel platform te respecteren, zoals verduidelijkt in 9.6.2.1 PLATFORMSPECIEKE USER EXPERIENCE. Deze vereiste was doorslaggevend bij de keuze van de geschikte oplossingsmethode in stadia 2 en 3.

Vervolgens werden de voornaamste use cases als voorbeelden uitgewerkt (zie bijlage 9.9.1 BESCHRIJVINGEN). Een use case omschrijft de stappen die een potentiële gebruiker van de app moet doorlopen om een functie uit te voeren. Het geeft samen met de requirements analyse een beeld van hoe de algemene gebruikerservaring er zal uitzien.

Om daarna een overzicht te krijgen van het parcours dat een gebruiker binnen de toepassing kan afleggen, werd een use case diagram opgesteld. Dit diagram is terug te vinden als bijlage 9.9.2 DIAGRAM. Het geeft weer wat alle mogelijke *schermen* die een gebruiker kan zien of acties die hij/zij kan nemen zijn. Hiervoor is de gestandaardiseerde manier om een use case diagram op te stellen gebruikt. De use case beschrijvingen en het use case diagram hebben als basis gediend voor de opbouw van het prototype in fase 4.

Tot slot werd alle informatie gebundeld in de vorm van een Project Initiatie Document (PID) dat terug te vinden is als bijlage 9.5 PROJECT INITIATIE DOCUMENT. Dit document dient als *contract* tussen de opdrachtgever (RealDolmen) en de uitvoerder van het project. In december keurden alle betrokken partijen het PID goed. Het bevat alle concrete afspraken over wat het project inhoudt. Daarnaast werd er ook een inschatting gemaakt van de risico's (9.5.5.4 RISICO'S) en werd er vastgelegd hoe de oplevering in zijn werk gaat (9.5.6 BEHEERSASPECTEN). RealDolmen gebruikt bijvoorbeeld een Microsoft Team Foundation Server om de broncode van alle projecten gecentraliseerd op te slaan. De risicoschatting kon pas tijdens fases 3 en 4 voltooid worden. Zo bleek de creatie van de mobiele applicatie voor iOS, het mobiele platform van Apple, moeilijker in praktijk dan in theorie. Hier wordt dan ook verder in dit document, bij stadia 3 en 4, op ingegaan.

²API: Application Programming Interface. Dit is een systeem dat gegevens beschikbaar stelt in een formaat dat door softwareontwikkelaars gebruikt kan worden om weer te geven in een applicatie of om te gebruiken bij de verwerking van dezelfde gegevens. Zie ook 9.10.1 ARCHITECTUUR.

Een van de belangrijkste onderdelen van een PID is de afbakening van het werk: de scope (9.5.4.4 REIKWIJDTE (SCOPE)). Dit concretiseert welke delen van het werk binnen het project vallen en welke niet. De ontwikkeling van een mobiele app impliceert namelijk vaak ook andere deliverables die zelf ook veel werk kosten. Zo heeft een applicatie nagenoeg altijd een API nodig, een kenmerkend visueel ontwerp en moet de app uiteindelijk ook gepubliceerd worden in een applicatiewinkel zoals de Apple App Store. Deze drie taken vallen in dit project buiten de scope, de focus van dit project ligt op de applicatie zelf.

4.3 Fase 2: onderzoek naar oplossingsmethodes

Nadat de huidige situatie en de vraag van de opdrachtgever in kaart gebracht zijn, vangt de zoektocht naar een goede oplossing aan. Voor elk probleem zijn er verschillende oplossingen. In fase 2 werden de oplossingen gezocht waarna ze in fase 3 ten opzichte van elkaar afgewogen werden.

In dit project was de kerntaak een systeem zoeken waarmee mobiele apps voor meerdere mobiele platformen ontwikkeld kunnen worden. Het is zonder meer namelijk onmogelijk om één applicatie te ontwikkelen die zowel op Google Android als op Apple iOS werkt. Om dit te verwezenlijken, bestaan er verscheidene commerciële oplossingen die bruggen bouwen tussen de mobiele platformen. Het blijft daarnaast mogelijk om alsnog aparte apps te maken voor elk mobiel platform. Dit laatste zou wel een enorme tijds kost met zich meebrengen. Een dergelijke oplossing wordt vaak een *cross-platform development environment* genoemd. Broncode wordt nagenoeg altijd geschreven aan de hand van software genaamd *integrated development environment (IDE)*. Een cross-platform IDE geeft dus aan dat er voor meerdere platformen mee ontwikkeld kan worden.

De meest courante cross-platform IDE's, inclusief het door RealDolmen zelf voorgestelde Xamarin, werden bestudeerd en gescoord op enkele eigenschappen. Zo is het belangrijk wat de kwaliteit is van de applicatie die afgeleverd wordt, wat de verhouding van shared code³ is en hoe eenvoudig het is voor een softwareontwikkelaar om met dit product te werken. Ook het gezamenlijk kunnen testen van de gegenereerde applicaties is een belangrijke eigenschap die de ontwikkeling ervan versoepelt. Het resultaat hiervan zijn de tabellen in bijlage 9.11 VERGELIJKING CROSS-PLATFORM DEVELOPMENT ENVIRONMENTS.

De eerste tabel bekijkt oppervlakkig wat de mogelijkheden zijn van elk product. Ook worden de ondersteunde mobiele platformen nagegaan. In de tweede tabel wordt de vraag van de klant afgetoetst tegenover elk mobiel platform. De belangrijkste eigenschappen zijn meer punten waard en irrelevante eigenschappen zijn niet langer vermeld.

Tijdens de opstelling van de tabellen met scores, werd vooral duidelijk dat cross-platform IDE's in te delen vallen in twee categorieën op basis van het type app dat ze afleveren: IDE's die web apps genereren en IDE's die native apps genereren. Daarom werd het meteen belangrijk om een vergelijkende analyse te doen van het onderscheid en de karakteristieken van beide types applicaties. Deze analyse is terug te vinden als bijlage 9.12 VERGELIJKING NATIVE EN WEB APPS.

Met de resultaten van de analyse kon er weer aan de slag gegaan worden om de oplossingen uit te zoeken die het beste met de vraag van de opdrachtgever overeen kwamen. Uit de requirements analyse was af te leiden dat de applicatie voor RealDolmen Education een native app moet zijn. Xamarin, de cross-platform IDE die ze zelf voorstelden te gebruiken, valt ook in deze categorie.

³ Het is onmogelijk om één mobiele app te bouwen die op meerdere mobiele platformen werkt. Er moet sowieso een aparte app gemaakt worden per mobiel platform. Dankzij enkele commerciële producten is het mogelijk om de broncode van deze apps met elkaar te delen zodat deze maar eenmalig geschreven moet worden. Hier wordt dus gekeken naar de verhouding gedeelde code ten opzichte van de broncode van alle apps samen. Wanneer meer dan de helft van de code gedeeld is, kunnen de verschillende apps per platform gezien worden als één app met enkele aanpassingen of optimalisaties per platform.

4.4 Fase 3: analyses van mogelijke oplossingen

Fase 3 ving in principe al aan tijdens de laatste stappen van fase 2. De vergelijking tussen native en web apps was een doorslaggevende analyse om de beste oplossing voor de klant te kunnen kiezen.

Het tweede stadium leidde het project verder naar Xamarin Inc., het bedrijf achter bekende cross-platform development producten. Xamarin biedt twee manieren aan om apps te ontwikkelen. De bekendste oplossing heet Xamarin⁴, maar een alternatief hiervoor is Xamarin.Forms. Beide oplossingen voldoen op het eerste zicht aan de vraag van de klant. Daarnaast is het ook mogelijk om de apps te ontwikkelen in zowel Xamarin Studio, een IDE van Xamarin als in Visual Studio, een IDE van Microsoft waar Xamarin mee geïntegreerd is.

Om de juiste aanpak voor implementatie vast te kunnen leggen, was een laatste onderzoek nodig. In bijlage 9.13 VERGELIJKING XAMARIN & XAMARIN.FORMS wordt dieper ingegaan op de verschillen tussen Xamarin (klassiek) en Xamarin.Forms.

Om deze analyse zo realistisch mogelijk op te stellen, werden ook meteen demo apps ontwikkeld aan de hand van Xamarin en Xamarin.Forms, terug te vinden via bijlage 9.16 DEMO APPS. Hiermee konden twee doelstellingen gehaald worden. Enerzijds was de ontwikkeling ervan, hoe kleinschalig ook, een rijke ervaring die bijgedragen heeft aan de eerder vermelde analyse en aan de inschatting van de risico's in het PID. Anderzijds maken ze ook deel uit van fase 4. Het werd namelijk eenvoudiger om een technische analyse op te stellen en de opgedane ervaring zal ongetwijfeld ook de implementatie van dit project ten goede komen.

Deze analyse bracht ook tekortkomingen met beide producten aan het licht. Xamarin ondersteunt verschillende platformen en gebruikt onderliggend een interessante architectuur om dit mogelijk te maken. Op platformen van Apple zit deze architectuur gecompliceerder in elkaar waardoor de ontwikkeling van de iOS-versie van een Xamarin app complexer is. Dit vormt een groot risico bij de ontwikkeling van elke Xamarin app en werd dan ook opgenomen in het PID.

Uit de conclusie van de analyse en Figuur 19 valt af te leiden dat de juiste keuze voor dit project naar de kant van Xamarin (klassiek) overhelt. Aan een implementatie met Xamarin.Forms is minder werk ten koste van een minder kwalitatieve gebruikerservaring per mobiel platform. Aangezien een degelijke ervaring per platform een vereiste van de opdrachtgever is (9.6.2.1 PLATFORMSPECIEKE USER EXPERIENCE), zal het project geïmplementeerd worden met Xamarin (klassiek). Deze variant van Xamarin is ook de variant die RealDolmen zelf voorgesteld heeft.

⁴ Hier wordt verder naar verwezen als *Xamarin* of als *Xamarin (klassiek)*.

4.5 Fase 4: voorbereiding van implementatie

Om het eerste luik van dit project, de voorbereiding, af te sluiten, moest de gekozen oplossing tot slot nog concreet voorbereid worden. Na de analyse in fase 3 kwam een Xamarin app naar voren als oplossing voor de vraag van de klant. De deliverables uit fase 3 zijn daarom op zich ook al voorbereidend werk voor de implementatie. Ook de documentatie uit stadium 1 (9.14 API DOCUMENTATIE) is waardevol als voorbereiding.

Vervolgens restten er nog twee uit te werken deliverables die de implementatie concretiseren. De eerste is een technische analyse die terug te vinden is als bijlage 9.10 TECHNISCHE ANALYSE.

4.5.1 Technische analyse

De technische analyse bestaat uit verschillende onderdelen. Om de ontwikkeling van een softwareproject te bespoedigen en duidelijk te structureren, wordt gebruik gemaakt van bestaande codebibliotheken. Deze bibliotheken – ook pakketten genoemd – bieden functies aan die vaak gebruikt worden zodat ze niet opnieuw opgebouwd moeten worden in elk softwareproject. In een project met Xamarin kunnen codebibliotheken aangewend worden die aangeboden worden als *NuGet package*. NuGet is een systeem om dergelijke pakketjes code gecentraliseerd aan te bieden en te verspreiden. De technische analyse bevat een lijst met voor dit project interessante NuGet packages.

Daarnaast bestaan er protocollen en standaarden die gevolgd en gerespecteerd dienen te worden bij de ontwikkeling van het project. Welke protocollen dit zijn, hoe ze werken en hoe ze geïmplementeerd kunnen worden, is ook in de technische analyse beschreven.

Tot slot beschrijft de technische analyse ook nog de architectuur van het project en hoe alle onderdelen met elkaar interageren.

4.5.2 Prototype

De tweede deliverable die dient om de implementatie te concretiseren is een prototype. Het prototype is in dit geval een verzameling van wireframes die weergeven hoe de applicatie visueel opgebouwd zal worden. Wireframes zijn blauwdrukken van alle schermen die een gebruiker kan zien en de stappen die hij moet doorlopen om deze te zien krijgen. De use case beschrijvingen en het use case diagram (9.9 USE CASES) uit fase 1 liggen aan de basis hiervan. Bijlage 9.15 SCHERMEN VAN WIREFRAMES bevat schermafdrucken van deze wireframes.

4.5.3 Evenementen

Tijdens het verloop van de 4 stadia van dit project vonden activiteiten plaats waarvan de inhoud een bijdrage kon leveren aan de uitkomst van het project. De evenementen waren telkens sessies georganiseerd door Belgische *user groups* van softwareontwikkelaars. Een user group bestaat uit enthousiastelingen die sessies geven waarbij nieuwe technologie, bekende problemen en oplossingen uit de doeken gedaan worden. De organisatoren zijn meestal vrijwilligers en de werking van de groep wordt gesponsord door bedrijven uit de sector waarin de groep actief is.

Van de drie bijgewoonde evenementen zijn verslagen te vinden als bijlage 9.4 VERSLAGEN EVENTS. Zo bevatte de sessie van MADN⁵ nuttige methodologieën om details van de technische analyse uit te werken. Vooral het protocol dat RealDolmen voor de authenticatie gebruikt, OAuth, werd tijdens die sessie gedemonstreerd.

⁵ MADN: Metro App Developer Network. Belgische user group van softwareontwikkelaars gespecialiseerd in de ontwikkeling van apps voor Microsoft Windows. Zie ook 9.4.3.1 WIE OF WAT IS MADN?.

4.5.4 Laatste voorbereiding

RealDolmen heeft een abonnement beschikbaar gesteld voor PluralSight. Dit is een online lessysteem waarbij gebruikers – tegen betaling – korte opleidingen van drie tot vijf uren kan volgen rond een bepaald onderwerp. Ook rond de ontwikkeling van Xamarin apps zijn er tientallen lessen beschikbaar.

Aan de hand van deze lessen en de gemaakte deliverables kan de voorbereiding van de implementatie afgewerkt worden.

5 Besluit

Na de vier voorbereidende fases volledig doorlopen te hebben, is dit project klaar voor implementatie en dat was ook het verwachte resultaat. Dit eerste deel is sterk bepalend geweest en heeft een duidelijk afgeleide context geschetst voor het verloop van de implementatie.

Het voornaamste doel is natuurlijk een waardevol eindresultaat na afloop van de stage: één Xamarin app die beschikbaar is voor Android, iOS en Windows Phone en volledig voldoet aan alle verwachtingen beschreven in de projectdefinitie. Op zich hebben de vier fases ook interessante deelproducten opgeleverd.

Zo zijn de drie vergelijkende analyses herbruikbaar voor andere projecten. Het is volstrekt mogelijk om elke business case met vraag naar een mobiele applicatie op meerdere platformen af te toetsen tegen elke analyse. De enige beperking is dat de vergelijkingen vooral doorgetrokken worden in de richting van native apps. Een bedrijf dat na de vergelijking tussen web en native apps moet concluderen dat een web/hybride app de beste oplossing is, moet zelf nog op zoek naar een oplossing.

Overigens is tijdens de opstelling van de analyses gebleken dat de beschikbare oplossingen op de markt nog sterk evolueren, waardoor de juiste keuze niet altijd meteen voor de hand ligt. Xamarin of Xamarin.Forms vormen voor veel mobiele app-projecten een gulden middenweg, maar alternatieve oplossingen bestaan momenteel nog niet. Uit deze vaststellingen kan besloten worden dat de ontwikkeling van applicaties voor meerdere platformen een recente trend is die nog volop in ontwikkeling is. Veel softwarebedrijven richten hun pijlen op deze nieuwe, te veroveren markt, maar behalve Xamarin slagen er maar weinigen in om een aantrekkelijke oplossing aan te bieden. Deze softwarebedrijven bevinden zich als het ware in een race om de mogelijkheden van hun rechtstreekse concurrenten zelf ook te kunnen aanbieden.

De gevolgen van deze situatie zijn voelbaar voor de softwareontwikkelaar die er uiteindelijk mee aan de slag gaat. De structuur van Xamarin voor iOS is dermate ingewikkeld waardoor een goede werking van veel factoren komt af te hangen. Er lijkt nog veel werk weggelegd voor zowel de ontwikkelaars van cross-platform IDE's als Xamarin Inc. als de ontwikkelaars van mobiele platformen zoals Google, Apple en Microsoft.

Deze laatste partijen komen steeds vaker voor een keuze te staan. Ofwel gaan ze elkaars mogelijkheden overnemen en zo cross-platform IDE's ondersteunen, ofwel bieden ze zelf het hoofd aan de grote vraag naar apps voor meerdere platformen en gaan ze dit zelf faciliteren. Ten slotte kunnen ze er ook voor kiezen om het eigen mobiele platform hiervan af te schermen om zo unieke mogelijkheden te bieden aan hun eigen klanten. Een status quo is geen optie aangezien elke beslissing voor de verdere ontwikkeling van mobiele platformen meteen een weerslag heeft op elke cross-platform IDE. Als Apple bijvoorbeeld met een nieuwe versie van iOS uit komt, dan is het aan Xamarin Inc. om meteen te reageren en ervoor te zorgen dat Xamarin voor iOS volledig compatibel is. Het gevolg kan dan een complexe(re) architectuur zijn zoals dat bij Xamarin voor iOS vandaag het geval is.

Niettemin is zelfs met alle kanttekeningen in het achterhoofd Xamarin op het moment van schrijven een degelijke oplossing voor een business case zoals die van RealDolmen Education. De implementatie vereist aandacht en voorbereiding wat net het doel was van dit project en van fase 4 in het bijzonder.

6 Zelfreflectie

Bij de zoektocht naar een project voor mijn bachelorproef en stage heb ik bewust gekozen voor een technologie waarmee ik zelf geen of beperkte ervaring had. Big data en cross-platform development zijn twee 'trends' die de laatste jaren mijn aandacht getrokken hebben. Rond big data waren maar weinig projecten beschikbaar waardoor de knoop snel doorgehakt was. Cross-platform development was daarentegen geen 'plan B', aangezien ik doorheen de vorige fases van de opleiding Toegepaste Informatica dit onderwerp ook altijd al interessant vond, zeker in combinatie met het mobiele gegeven. De keuze voor een project bij Real-Dolmen was logisch aangezien ik er kennissen heb, de locatie ideaal is voor mezelf en het bedrijf sterk staat in de Belgische IT-sector.

Na afloop van dit eerste deel, ben ik tevreden met die beslissing. Er zit duidelijk toekomst in cross-platform development en de grote ervaring die ik hieromtrent opgedaan heb is zeer waardevol. Zonder er effectief mee te werken had ik niet geweten dat de installatie van Xamarin voor iOS een complex gebeuren is dat niet altijd vlekkeloos verloopt. Ook de architectuur die Xamarin onderliggend gebruikt is heel interessant om eens te bestuderen. De analyses hebben zeker bijgedragen tot mijn kennis rond softwareontwikkeling en de werking van software in het algemeen.

Voor ik aan dit project begon zag ik mezelf als een *developer*⁶ in de meeste domeinen van softwareontwikkeling. Na afloop van dit eerste deel evolueert dit eerder in de richting van *technical specialist*⁶. Dankzij de projectmatige aanpak en de uitgebreide projectdefinitie heb ik volgens mijn eigen mening ook enkele vaardigheden van een *project manager*⁶ verworven.

Zo zijn er dan ook competenties die bij deze bachelorproef naar de oppervlakte gekomen zijn. Het zelfstandig aanleren en hanteren van nieuwe technologieën en principes is natuurlijk de voornaamste, maar ook het correct kunnen omschrijven van de vraag van de klant is een competentie die ik in dit project verder ontwikkeld heb. Beide competenties dragen bij aan de benodigde vaardigheden om aan het werk te kunnen als ICT consultant.

Tot slot heb ik een goed gevoel bij de stage. Alhoewel de indruk kan bestaan alsof het 'echte' werk nog moet beginnen, is dit volgens mij niet zo. De voorbereiding is verlopen zoals gepland en de tussentijdse feedback heb ik steeds ter harte genomen. Hierdoor sta ik naar mijn mening sterker in mijn schoenen om een vlotte start te nemen en het project steeds onder controle te houden. Ik mis nog ervaring om van mezelf te kunnen zeggen dat ik nu aan de slag kan als Xamarin app ontwikkelaar, maar die verwacht ik tijdens de stage op te doen.

⁶ Profiel uit het E-Competence Framework, een Europese referentie om ICT-vaardigheden en -profielen te kaderen.

7 Figurenlijst

Figuur 1: logo van RealDolmen.....	8
Figuur 2: de website van RealDolmen Education.....	10
Figuur 3: fases van dit project.....	11
Figuur 4: aantal correspondenties per maand	29
Figuur 5: gespendeerde tijd per maand.....	32
Figuur 6: algemeen overzicht van werking van mobile clients.....	66
Figuur 7: overzicht van API-modellen	69
Figuur 8: indeling van scopes en claims	70
Figuur 9: authenticatie en autorisatie	70
Figuur 10: verloop van authenticatie en autorisatie	71
Figuur 11: Proces van compileren van programmacode met voorbeelden	75
Figuur 12: Proces van uitvoering van native code met voorbeelden.....	76
Figuur 13: opzet van Xamarin	80
Figuur 14: van applicatiecode naar oproepen in het Android-systeem.....	82
Figuur 15: ontwikkelproces met Xamarin.iOS	83
Figuur 16: ontwikkelervaring met Xamarin.iOS.....	84
Figuur 17: structuur van een project met apps voor meerdere platformen en PCL's	84
Figuur 18: structuur van Xamarin.Forms	85
Figuur 19: criteria Xamarin / Xamarin.Forms	87

8 Bibliografie

Adobe Systems Inc. (2015). *About the Project*. Opgehaald van PhoneGap: <http://phonegap.com/about/>

Adobe Systems Inc. (2015). *Supported Features*. Opgehaald van PhoneGap: <http://phonegap.com/about/feature/>

Appcelerator Inc. (2015). *Mobile App Development Platform*. Opgehaald van Appcelerator: <http://www.appcelerator.com/>

Appcelerator Inc. (2015). *Products*. Opgehaald van Appcelerator: <http://www.appcelerator.com/product/>

AZUG. (2015, juni 23). *2015-08-22 - 3 sessions and a BBQ*. Opgehaald van AZUG: <http://www.azug.be/events/2015-08-22---3-sessions-and-a-bbq>

AZUG. (2015, september 3). *AzureCon Viewing Party*. Opgehaald van AZUG: <http://www.azug.be/events/2015-09-29---azurecon-viewing-party>

Budiu, R. (2013, september 14). *Mobile: Native Apps, Web Apps, and Hybrid Apps*. Opgehaald van Nielsen Norman Group: <http://www.nngroup.com/articles/mobile-native-apps/>

Cabus, W. (2015). Securing and consuming a Web API with OAuth. *MADN*. Kontich: RealDolmen. Opgeroepen op december 18, 2015, van <https://www.eventbrite.com/e/securing-and-consuming-a-web-api-with-oauth-tickets-19726330987>

Dockx, K. (2015, maart 5). Building and Securing a RESTful API for Multiple Clients in ASP.NET. Pluralsight. Opgehaald van <https://app.pluralsight.com/library/courses/building-securing-restful-api-aspdotnet/>

Fielding, R. T. (2000). Representational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures*. Irvine, United States of America: University of California. Opgehaald van http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Google Inc. (2016, januari 15). *xamarin - phonegap - apache cordova*. Opgehaald van Google Trends: <https://www.google.be/trends/explore#q=xamarin%2C%20phonegap%2C%20apache%20cordova&cmpt=q&tz=Etc%2FGMT-1>

Google Inc. (sd). *ART and Dalvik*. Opgehaald van Android: <https://source.android.com/devices/tech/dalvik/index.html>

Google Inc. (sd). *V8 Javascript Engine*. Opgehaald van Google Code: <http://code.google.com/p/v8/>

Gregory, K. (2003, april 28). *Managed, Unmanaged, Native: What Kind of Code Is This?* Opgehaald van developer.com: <http://www.developer.com/net/cplus/article.php/2197621/Managed-Unmanaged-Native-What-Kind-of-Code-Is-This.htm>

Jobs, S. (2010, april). *Thoughts on Flash*. Opgehaald van Apple: <http://www.apple.com/hotnews/thoughts-on-flash/>

Microsoft Corp. (2015, september). *AzureCon*. Opgehaald van Microsoft Azure: <https://azure.microsoft.com/en-us/azurecon/>

- Microsoft NV/SA. (2015, september). *AzureConf View Party - together with AZUG*. Opgehaald van Microsoft Events: <https://msevents.microsoft.com/CUI/EventDetail.aspx?culture=en-BE&eventid=1032664866&flag=3>
- Mozilla Contributors. (2015, mei 12). *Rhino*. Opgehaald van Mozilla Developer Network: <https://developer.mozilla.org/nl/docs/Mozilla/Projects/Rhino>
- Mudge, J. T. (2012, juli 11). *Native App vs. Mobile Web App: A Quick Comparison*. Opgehaald van Six Revisions: <http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/>
- Nabar, S. (2015, november 24). *Demystifying HttpClient APIs in the Universal Windows Platform*. Opgehaald van Windows blogs: <http://blogs.windows.com/buildingapps/2015/11/23/demystifying-httpclient-apis-in-the-universal-windows-platform/>
- Odisee UC. (2015, september). *Competentieprofielen / Opleidingsspecifieke leerresultaten (OLR) academiejaar 2015-2016*. Opgehaald van Odisee: <https://webapps.odisee.be/ECTSCompetenties/Competentieprofiel.aspx?taal=N&OPLID=51&ACJ=2015>
- RealDolmen NV. (2012, februari 7). *RealDolmen logo*. Opgehaald van <http://www.realdolmen.com/static/images/RealDolmen-Logos/logo.png>
- RealDolmen. (sd). *Opleidingen*. Opgehaald van RealDolmen: <http://realdolmen.be/nl/education/about-us>
- RealDolmen. (sd). *Welkom bij RealDolmen Education*. Opgehaald van RealDolmen Education: <https://education.realdolmen.com/nl>
- The Apache Software Foundation. (2015, oktober). *Overview*. Opgehaald van Apache Cordova: <https://cordova.apache.org/docs/en/5.1.1/guide/overview/index.html>
- The Apache Software Foundation. (2015, oktober). *Platform Support*. Opgehaald van Apache Cordova: <https://cordova.apache.org/docs/en/5.1.1/guide/support/index.html>
- W3C. (sd). *HTML Working Group*. Opgehaald van W3C: <http://www.w3.org/html/wg/>
- Wikipedia contributors. (2015, mei 1). *Bytecode*. Opgehaald van Wikipedia, The Free Encyclopedia.: <https://en.wikipedia.org/w/index.php?title=Bytecode&oldid=660227726>
- Wikipedia contributors. (2015, september 22). *Comparison of instruction set architectures*. Opgehaald van Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/w/index.php?title=Comparison_of_instruction_set_architectures&oldid=682216410
- Wikipedia contributors. (2015, november 17). *IOS*. Opgehaald van Wikipedia, The Free Encyclopedia: <https://en.wikipedia.org/w/index.php?title=IOS&oldid=691147548>
- Wilde, E. (2013, april 13). *How does Xamarin.iOS aka monotouch work?* Opgehaald van Edward Wilde: <http://blogs.edwardwilde.com/2013/04/13/how-does-xamarin-ios-aka-monotouch-work/>
- Xamarin Inc. (2015). *An Introduction to Xamarin.Forms*. Opgehaald van Xamarin Developers: <http://developer.xamarin.com/guides/cross-platform/xamarin-forms/getting-started/introduction-to-xamarin-forms/>

Xamarin Inc. (2015). *Cross-Platform User Interfaces with Xamarin.Forms*. Opgehaald van Xamarin Developers: <http://developer.xamarin.com/guides/cross-platform/xamarin-forms/>

Xamarin Inc. (2015). *Introduction to Portable Class Libraries*. Opgehaald van Xamarin Developers: http://developer.xamarin.com/guides/cross-platform/application_fundamentals/pcl/introduction_to_portable_class_libraries/

Xamarin Inc. (2015). *Portable Class Libraries*. Opgehaald van Xamarin Developers: http://developer.xamarin.com/guides/cross-platform/application_fundamentals/pcl/

Xamarin Inc. (2015). *Sharing Code Options*. Opgehaald van Xamarin Developers: http://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/sharing_code_options/

Xamarin Inc. (2015). *Xamarin Platform*. Opgehaald van Xamarin: <https://xamarin.com/platform>

Xamarin Inc. (2015). *Xamarin.Forms*. Opgehaald van Xamarin: <https://xamarin.com/forms>

Xamarin Inc. (sd). *API Design*. Opgehaald van Xamarin Developers: https://developer.xamarin.com/guides/ios/under_the_hood/api_design/

Xamarin Inc. (sd). *Async Support Overview*. Opgehaald van Xamarin: http://developer.xamarin.com/guides/cross-platform/advanced/async_support_overview/

Xamarin Inc. (sd). *Limitations*. Opgehaald van Xamarin Developers: http://developer.xamarin.com/guides/ios/advanced_topics/limitations/

9 Bijlagen

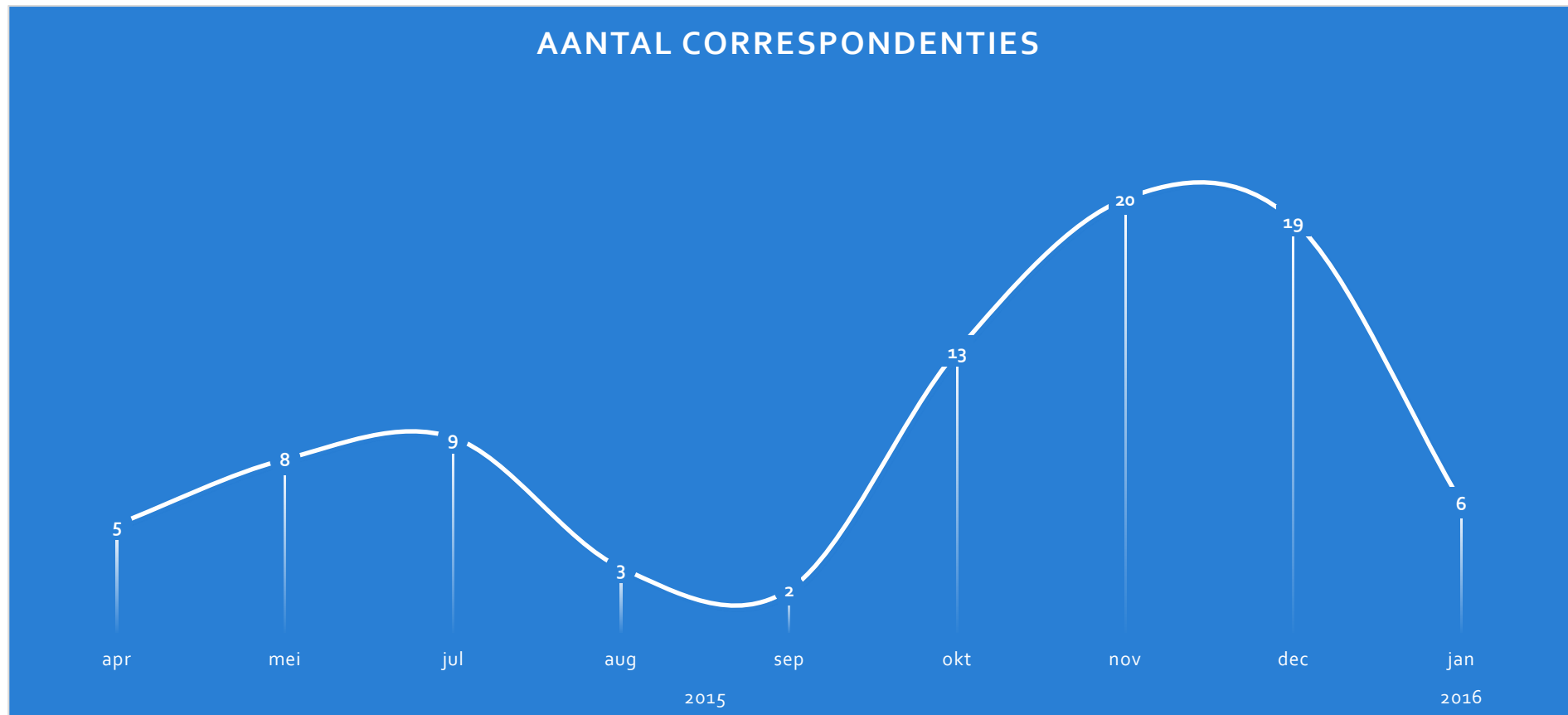
9.1 Contacthistoriek

Datum	Verzender	Ontvanger	Bedrijf	Type	Onderwerp
do 2 apr	Sam	Annika Beeck	SAS	mondeling	Vraag voor mogelijkheid stage bij SAS, zochten geen technische profielen voor stages.
do 2 apr	Sam	Lieve Goedhuys	Delaware	mondeling	Tijdens job-dag SAS, vraag voor mogelijkheid big data-stage. LG ging zelf mail sturen.
do 2 apr	Sam	Vanessa Closset	Keyrus	mondeling	Tijdens job-dag SAS, vraag voor mogelijkheid big data-stage. VC ging zelf mail sturen.
vr 24 apr	Sam	Wouter Thielemans	Odisee	mondeling	Opzoekwerk mogelijke stageplaatsen big data
vr 24 apr	Wouter Thielemans	Sam	Odisee	e-mail	Lijst mogelijke stageplaatsen big data
za 2 mei	Sam	Wouter Thielemans	Odisee	e-mail	Vraag naar contactpersonen RealDolmen en Keyrus
ma 4 mei	Wouter Thielemans	Sam	Odisee	e-mail	Contactpersoon Keyrus
vr 15 mei	Wouter Thielemans	Lieve Goedhuys	Delaware	mondeling	Vraag of er nieuws is over big data-stage
zo 17 mei	Wouter Thielemans	Sam	Odisee	e-mail	Contactpersoon RealDolmen big data
di 19 mei	Sam	Lieve Goedhuys	Delaware	e-mail	Sollicitatie big data-stage
di 19 mei	Sam	Erik De Romagnoli	RealDolmen	e-mail	Sollicitatie big data-stage
wo 20 mei	Erik De Romagnoli	Sam	RealDolmen	e-mail	Gaat kijken of er in RealDolmen mogelijkheid is voor big data-stage
zo 24 mei	Sam	Lieve Goedhuys	Delaware	LinkedIn	Vraag of er nieuws is over big data-stage
di 14 jul	Sam	n.v.t.	Euricom	e-mail	Sollicitatie .NET-stage
ma 20 jul	Erik De Romagnoli	Sam	RealDolmen	e-mail	Geen big data-stages
vr 24 jul	Sam	n.v.t.	Euricom	telefonisch	Vraag naar update sollicitatie. Euricom ging de maandag erna terugbellen.
wo 29 jul	Sam	n.v.t.	Euricom	telefonisch	Vraag naar update sollicitatie. Geen mogelijkheid voor stage.
do 30 jul	Sam	Wesley Cabus	RealDolmen	Twitter	Vraag over wie stages voor .NET regelt. Geeft mscommunity@real-dolmen.be door.
do 30 jul	Sam	MS Community	RealDolmen	e-mail	Sollicitatie .NET-stage
do 30 jul	Wesley Cabus	Sam	RealDolmen	Twitter	Bevestiging ontvangst sollicitatie
vr 31 jul	Alexander Reynaert	Sam	RealDolmen	e-mail	Stageprojecten .NET
vr 31 jul	Erik De Romagnoli	Sam	RealDolmen	telefonisch	Afspraak sollicitatie
wo 12 aug	Sam	Erik De Romagnoli & Alexander Reynaert	RealDolmen	mondeling	Sollicitatie

wo 12 aug	Nico Vermeir	Sam	RealDolmen	e-mail	Stageprojecten .NET
wo 12 aug	Sam	Nico Vermeir	RealDolmen	e-mail	Keuze stageproject
do 24 sep	Sam	Nico Vermeir	RealDolmen	e-mail	Vastleggen afspraak
vr 25 sep	Nico Vermeir	Sam	RealDolmen	e-mail	Bevestiging afspraak op 2/10
vr 2 okt	Sam	Nico Vermeir	RealDolmen	mondeling	Overlopen van project, scope
vr 2 okt	Nico Vermeir	Sam	RealDolmen	e-mail	Broncode van API die bij de applicatie gebruikt moet worden
vr 2 okt	Sam	Nico Vermeir	RealDolmen	e-mail	Handleiding van de bachelorproef
zo 4 okt	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag over enkele requirements van het project
zo 4 okt	Nico Vermeir	Sam	RealDolmen	e-mail	Verduidelijking projectomschrijving
zo 25 okt	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag voor afspraak om deliverables te bespreken
ma 26 okt	Nico Vermeir	Sam	RealDolmen	e-mail	Bevestiging afspraak om deliverables te bespreken
ma 26 okt	Sam	Nico Vermeir	RealDolmen	e-mail	Uur voor afspraak deliverables
ma 26 okt	Nico Vermeir	Sam	RealDolmen	e-mail	Bevestiging uur afspraak
ma 26 okt	Sam	Nico Vermeir	RealDolmen	e-mail	Bevestiging uur afspraak
ma 26 okt	Sam	Serge Van Cleynenbreugel	Odisee	mondeling	Vraag voor afspraak om voortgang project te bespreken
vr 30 okt	Sam	Nico Vermeir	RealDolmen	mondeling	Bespreking deliverables
za 31 okt	Sam	Gert Cauwenbergh	Odisee	e-mail	Vraag voor feedback over vorm van PID
ma 2 nov	Sam	Nico Vermeir	RealDolmen	e-mail	Vragen over use cases en technische analyse
ma 9 nov	Sam	Gert Cauwenbergh	Odisee	mondeling	Vraag over feedbackmail
ma 9 nov	Gert Cauwenbergh	Sam	Odisee	e-mail	Vraag over feedbackmail
ma 9 nov	Serge Van Cleynenbreugel	Sam	Odisee	mondeling	Feedback over voortgang project
di 10 nov	Sam	Gert Cauwenbergh	Odisee	e-mail	Verduidelijking bij feedbackmail
vr 13 nov	Nico Vermeir	Sam	RealDolmen	e-mail	Antwoord over use cases
vr 13 nov	Gert Cauwenbergh	Sam	Odisee	e-mail	Feedback over PID en Planning
ma 16 nov	Nico Vermeir	Sam	RealDolmen	e-mail	Antwoord over technische analyse
ma 16 nov	Erik De Romagnoli	Sam	RealDolmen	e-mail	Vraag over stagecontract
di 17 nov	Sam	Yvan Rooseleer & Chantal Teerlinck	Odisee	e-mail	Vraag over stagecontract
di 17 nov	Sam	Erik De Romagnoli	RealDolmen	telefonisch	Antwoord over stagecontract
za 21 nov	Sam	Serge Van Cleynenbreugel & Nico Vermeir	Odisee & RealDolmen	e-mail	Vraag voor feedback over web/native vergelijking
wo 25 nov	Sam	Nico Vermeir	RealDolmen	Twitter	Vraag over requirements

wo 25 nov	Nico Vermeir	Sam	RealDolmen	Twitter	Antwoord over requirements
do 26 nov	Sam	Serge Van Cleynenbreugel & Nico Vermeir	Odisee & RealDolmen	e-mail	Uitnodiging infosessie stage
za 28 nov	Serge Van Cleynenbreugel	Sam	Odisee	e-mail	Feedback over web/native vergelijking
zo 29 nov	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag voor feedback over web/native vergelijking & Xamarin.Forms
zo 29 nov	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag over delen broncode
ma 30 nov	Nico Vermeir	Sam	RealDolmen	e-mail	Antwoord over delen broncode
ma 30 nov	Nico Vermeir	Sam	RealDolmen	e-mail	Antwoord uitnodiging infosessie
di 1 dec	Sam	Nico Vermeir	RealDolmen	e-mail	Antwoord uitnodiging infosessie
di 1 dec	Nico Vermeir	Sam	RealDolmen	e-mail	Feedback vergelijking web/native
di 1 dec	Nico Vermeir	Sam	RealDolmen	e-mail	Feedback Xamarin demo
di 1 dec	Sam	Nico Vermeir	RealDolmen	e-mail	Feedback Xamarin demo
di 1 dec	Nico Vermeir	Sam	RealDolmen	e-mail	Feedback Xamarin demo
di 1 dec	Nico Vermeir	Sam	RealDolmen	e-mail	Feedback Xamarin demo
wo 2 dec	Sam	Yvan Rooseleer & Chantal Teerlinck	Odisee	e-mail	Herinnering vraag over stagecontract
wo 2 dec	Yvan Rooseleer	Sam	Odisee	e-mail	Antwoord vraag stagecontract
do 3 dec	Chantal Teerlinck	Sam	Odisee	e-mail	Antwoord vraag stagecontract
vr 4 dec	Sam	Serge Van Cleynenbreugel & Nico Vermeir	Odisee & RealDolmen	e-mail	Vraag feedback vergelijking Xamarin / Xamarin.Forms
vr 4 dec	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag over API
ma 7 dec	Sam	Erik De Romagnoli	RealDolmen	e-mail	Antwoord gegevens stage
ma 7 dec	Sam	Kevin Dockx	RealDolmen	e-mail	Vraag over OAuth API
ma 7 dec	Nico Vermeir	Sam	RealDolmen	e-mail	Antwoord over vraag API
ma 7 dec	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag goedkeuring PID
vr 11 dec	Sam	Nico Vermeir	RealDolmen	e-mail	Bijgewerkt PID
di 15 dec	Nico Vermeir	Sam	RealDolmen	e-mail	Goedgekeurd PID
wo 30 dec	Sam	Serge Van Cleynenbreugel	Odisee	e-mail	Vraag feedback vergelijking Xamarin / Xamarin.Forms
wo 30 dec	Sam	Nico Vermeir	RealDolmen	e-mail	Vraag feedback vergelijking Xamarin / Xamarin.Forms
ma 4 jan	Serge Van Cleynenbreugel	Sam	Odisee	e-mail	Feedback Xamarin / Xamarin.Forms
ma 4 jan	Sam	Serge Van Cleynenbreugel	Odisee	e-mail	Vraag feedback structuur integratietekst
di 5 jan	Serge Van Cleynenbreugel	Sam	Odisee	e-mail	Feedback structuur integratietekst

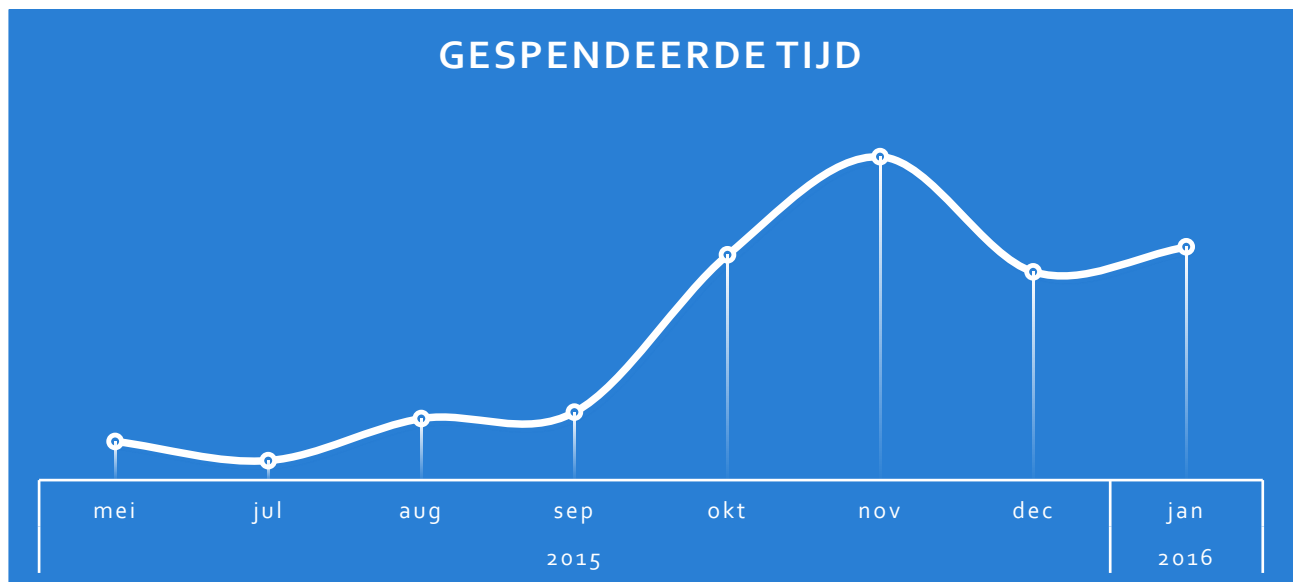
di 5 jan	Sam	Serge Van Cleynenbreugel & Nico Vermeir	Odisee & RealDolmen	e-mail	Uitnodiging presentatie bachelorproef
di 5 jan	Nico Vermeir	Sam	RealDolmen	e-mail	Antwoord uitnodiging presentatie
do 14 jan	Nico Vermeir	Sam	RealDolmen	e-mail	Feedback Xamarin / Xamarin.Forms

*Figuur 4: aantal correspondenties per maand*

9.2 Tijdsbesteding

Omschrijving	Startdatum	Starttijdstip	Einddatum	Eindtijdstip	Duur
Motivatiefbrief Delaware schrijven	19/05/2015	10:00:00	19/05/2015	12:00:00	02:00:00
CV opstellen	19/05/2015	14:00:00	19/05/2015	17:00:00	03:00:00
Motivatiefbrief Euricom schrijven	14/07/2015	10:00:00	14/07/2015	11:30:00	01:30:00
Motivatiefbrief RealDolmen schrijven	30/07/2015	10:00:00	30/07/2015	11:00:00	01:00:00
Sollicitatie stageproject	12/08/2015	10:00:00	12/08/2015	10:45:00	00:45:00
Event: AZUG	22/08/2015	8:45:00	22/08/2015	16:00:00	07:15:00
Contacthistoriek bijwerken	26/09/2015	14:00:00	26/09/2015	17:00:00	03:00:00
Les IT-project	28/09/2015	14:00:00	28/09/2015	15:50:00	01:50:00
Event: AZUG	29/09/2015	17:30:00	29/09/2015	21:30:00	04:00:00
Gesprek met Nico over stageproject	2/10/2015	11:30:00	2/10/2015	12:00:00	00:30:00
Vergelijking features ontwikkelomgevingen	4/10/2015	13:00:00	4/10/2015	15:40:00	02:40:00
Projectomschrijving (PPT)	4/10/2015	15:40:00	4/10/2015	16:05:00	00:25:00
Vergelijking features ontwikkelomgevingen	4/10/2015	16:15:00	4/10/2015	17:10:00	00:55:00
Projectomschrijving (Word)	4/10/2015	21:02:00	4/10/2015	21:54:00	00:52:00
Contacthistoriek bijwerken	4/10/2015	22:15:00	4/10/2015	22:30:00	00:15:00
Installatie Hyper-V (voor emulators)	11/10/2015	22:10:00	11/10/2015	23:13:00	01:03:00
Les IT-project	12/10/2015	14:00:00	12/10/2015	16:00:00	02:00:00
Vorbereiding presentatie (pitch)	16/10/2015	14:19:00	16/10/2015	15:35:00	01:16:00
Projectomschrijving / lijst deliverables opstellen	17/10/2015	14:20:30	17/10/2015	14:46:18	00:25:48
Les IT-Project	19/10/2015	14:00:00	19/10/2015	16:20:00	02:20:00
Lijst deliverables afwerken	24/10/2015	22:03:06	24/10/2015	22:31:12	00:28:06
Project Planning	24/10/2015	22:50:52	24/10/2015	23:12:06	00:21:14
Bibliografie (tussentijds) oplijsten	24/10/2015	23:47:04	24/10/2015	23:53:15	00:06:11
Bibliografie (tussentijds) oplijsten	25/10/2015	12:02:34	25/10/2015	12:50:56	00:48:21
Verslagen AZUG events	25/10/2015	13:51:00	25/10/2015	17:10:00	03:19:00
PID opstellen	25/10/2015	18:38:25	25/10/2015	19:06:22	00:27:57
PID opstellen	25/10/2015	23:19:00	26/10/2015	1:27:00	02:08:00
PID versie 2	26/10/2015	13:13:00	26/10/2015	13:40:00	00:27:00
Les IT-Project	26/10/2015	14:00:00	26/10/2015	16:00:00	02:00:00
Gesprek met Nico over deliverables	30/10/2015	12:00:00	30/10/2015	12:30:00	00:30:00
Feedback afgelopen week verwerken	30/10/2015	19:50:11	30/10/2015	22:07:46	02:17:36
PID versie 2	31/10/2015	13:37:00	31/10/2015	14:45:55	01:08:55
Presentatie deliverables	31/10/2015	15:23:22	31/10/2015	16:14:17	00:50:55
Bestaande API onderzoeken	31/10/2015	19:25:00	31/10/2015	21:14:00	01:49:00
Bestaande API onderzoeken	1/11/2015	11:51:22	1/11/2015	12:31:36	00:40:15
Use cases	2/11/2015	18:55:00	2/11/2015	19:29:00	00:34:00
Use cases	3/11/2015	19:55:00	3/11/2015	21:35:00	01:40:00
Use cases	8/11/2015	17:48:00	8/11/2015	18:15:00	00:27:00
Bespreking voortgang met SVC	9/11/2015	12:55:00	9/11/2015	13:35:00	00:40:00
Les IT-project	16/11/2015	14:00:00	16/11/2015	16:00:00	02:00:00
Mails e.d. afhandelen	17/11/2015	15:28:00	17/11/2015	15:50:00	00:22:00
Verwerking feedback	17/11/2015	22:33:21	17/11/2015	23:28:24	00:55:02
Vergelijking native/web apps	19/11/2015	22:08:34	19/11/2015	23:54:11	01:45:37
Vergelijking native/web apps	20/11/2015	14:20:00	20/11/2015	16:00:00	01:40:00

Vergelijking native/web apps	21/11/2015	13:13:09	21/11/2015	14:31:54	01:18:44
Xamarin research	24/11/2015	19:01:29	24/11/2015	23:31:15	04:29:47
Xamarin research	26/11/2015	23:54:57	27/11/2015	1:13:13	01:18:16
Xamarin research	27/11/2015	16:46:37	27/11/2015	18:00:57	01:14:20
Xamarin demo projects	27/11/2015	21:00:00	27/11/2015	22:07:00	01:07:00
Feedback vergelijking web/native verwerken	28/11/2015	14:45:53	28/11/2015	15:58:33	01:12:41
Xamarin demo projects	28/11/2015	16:00:47	29/11/2015	0:07:47	08:07:00
Xamarin demo projects	29/11/2015	12:03:00	29/11/2015	14:30:00	02:27:00
Vergelijking Xamarin / Xamarin.Forms	29/11/2015	14:47:57	29/11/2015	15:49:06	01:01:09
Vergelijking Xamarin / Xamarin.Forms	29/11/2015	16:54:00	29/11/2015	18:44:00	01:50:00
Vergelijking Xamarin / Xamarin.Forms	29/11/2015	21:40:37	29/11/2015	23:42:00	02:01:23
Vergelijking Xamarin / Xamarin.Forms	30/11/2015	9:06:00	30/11/2015	12:58:00	03:52:00
Les IT-project	30/11/2015	14:30:00	30/11/2015	16:00:00	01:30:00
Verwerking feedback	4/12/2015	15:23:35	4/12/2015	16:09:26	00:45:51
Technische analyse	4/12/2015	17:10:00	4/12/2015	17:50:00	00:40:00
Stagecontract opstellen	4/12/2015	18:30:00	4/12/2015	19:10:00	00:40:00
Technische analyse	4/12/2015	21:19:16	4/12/2015	22:19:40	01:00:24
Technische analyse	6/12/2015	22:00:00	6/12/2015	23:35:00	01:35:00
Les IT-project	7/12/2015	14:00:00	7/12/2015	16:00:00	02:00:00
Technische analyse	8/12/2015	20:30:00	8/12/2015	23:30:00	03:00:00
Infosessie stage	10/12/2015	18:00:00	10/12/2015	19:00:00	01:00:00
Spelling- en grammarverbeteringen aan afgewerkte deliverables	11/12/2015	20:55:19	11/12/2015	22:18:58	01:23:39
Voorbeelddesigns	11/12/2015	22:51:00	11/12/2015	23:21:30	00:30:30
Voorbeelddesigns	12/12/2015	21:45:00	12/12/2015	23:03:00	01:18:00
Voorbeelddesigns	13/12/2015	16:35:17	13/12/2015	18:24:18	01:49:00
Voorbeelddesigns	13/12/2015	21:26:05	13/12/2015	21:54:59	00:28:53
MADN: securing web API	15/12/2015	18:00:00	15/12/2015	21:00:00	03:00:00
Verslag MADN	18/12/2015	21:00:00	18/12/2015	22:00:00	01:00:00
Integratietekst	27/12/2015	21:06:00	28/12/2015	0:02:00	02:56:00
Integratietekst	30/12/2015	17:28:07	30/12/2015	18:20:22	00:52:15
Integratietekst	30/12/2015	20:56:12	30/12/2015	23:53:46	02:57:34
Integratietekst	31/12/2015	11:12:28	31/12/2015	11:25:19	00:12:50
Integratietekst	1/01/2016	13:24:00	1/01/2016	14:24:00	01:00:00
Integratietekst	2/01/2016	14:29:48	2/01/2016	15:10:13	00:40:25
Integratietekst	2/01/2016	15:56:20	2/01/2016	16:48:28	00:52:08
Integratietekst	3/01/2016	19:00:00	3/01/2016	23:06:00	04:06:00
Integratietekst	4/01/2016	18:11:24	4/01/2016	18:38:24	00:26:59
Integratietekst	4/01/2016	20:17:00	4/01/2016	22:20:00	02:03:00
Integratietekst	7/01/2016	20:47:00	7/01/2016	22:02:04	01:15:04
Integratietekst	9/01/2016	21:49:31	9/01/2016	23:01:01	01:11:31
Tips van KUL-tool	11/01/2016	15:59:00	11/01/2016	18:44:00	02:45:00
Tips van KUL-tool	12/01/2016	21:19:02	12/01/2016	23:50:57	02:31:55
Tips van KUL-tool	13/01/2016	17:28:38	13/01/2016	21:46:04	04:17:26
Integratietekst	14/01/2016	16:02:02	14/01/2016	16:55:10	00:53:07
Presentatie	14/01/2016	19:23:00	15/01/2016	1:20:00	05:57:00
Presentatie	15/01/2016	17:45:00	15/01/2016	20:11:00	02:26:00



Figuur 5: gespendeerde tijd per maand

9.3 Presentaties

9.3.1 Presentatie 1: pitch



Wie is de klant?

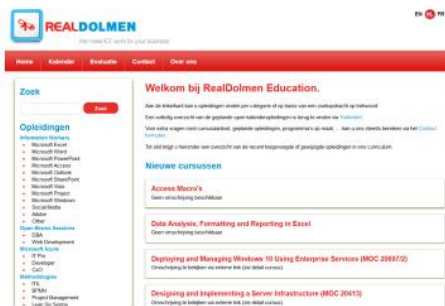
- RealDolmen: een van de grotere IT integrators in België
- RealDolmen Education: biedt talloze cursussen aan
- Opleidingen door RealDolmen zelf
- Mogelijkheid tot certificatie

Samuel Debruyne

15/01/2015

2

Situatie vandaag



Samuel Debruyne

15/01/2015

3

Vraag van de klant

- Mobiele app voor meerdere platformen
 - Android, iOS, Windows Phone
 - Zelfde mogelijkheden als op de website
 - Huisstijl van RealDolmen
 - Snel en flexibel
 - Voorstel: Xamarin (zie verder)

Samuel Debruyne

15/01/2015

4

5

Stageprojecten 2014-2015

- API met gegevens ontwikkeld (out of scope)
- Mobiele apps ontwikkeld in Xamarin.Forms (zie verder) – klant achteraf niet tevreden (in scope)

Scope (in / out of scope)

- | | |
|--|---|
| <ul style="list-style-type: none"> • 1 app voor Android, iOS, Windows Phone • Testen (e.v.t. simulator) • Design (visueel) • Engels, Frans, Nederlands | <ul style="list-style-type: none"> • API • Publicatie • Huisstijl • Hardware/software voor ontwikkeling |
|--|---|

Samuel Debruyne

15/01/2015

5

Samuel Debruyne

15/01/2015

6

Uitdagingen

- Duidelijk, gestructureerd
- 1 app, meerdere platformen
- Meertalig

Succes: SMART

- 1 app
- 3 platformen
- 3 talen
- 50% blauw/rood
- Alle opleidingen en onderverdelingen
- Zoekfunctie

Samuel Debruyne

15/10/2015

7

Samuel Debruyne

15/10/2015

8

odisee



Bedankt voor uw aandacht!

Samuel Debruyne

15/10/2015

ASSOCIATIE
KU LEUVEN

9

9.3.2 Presentatie 2: deliverables

Odisee



Het project



- RealDolmen Education: biedt talloze cursussen aan
- Mobiele app voor meerdere platformen
- Aanbod aan opleidingen weergeven



Persoonlijke deliverables

- CV
- Motivatiebrieven
- Contacthistoriek
- Bibliografie

Projectmanagement

- Time tracker: toggl.com
- Project Planning



Fase 1: projectdefinitie

- Requirements
 - Functioneel
 - Niet-functioneel
- Use cases

Fase 2: opstellen projectomschrijving

- Project Initiatie Document
- Project Closure Document

Fase 3: analyse van aanpak (1)

- Vergelijking cross platform development environments
- Vergelijkende analyse native / web apps
- Vergelijkende analyse Xamarin / Xamarin.Forms



Fase 3: analyse van aanpak (2)

- Technische analyse
- Verslagen events & lessen
 - Events van VISUG & AZUG
 - Lessen over cross platform development (m.b.v. Xamarin)



Fase 4: voorbereiding van project

- API in kaart brengen
- Voorbeeld in Xamarin.Forms
- Voorbeeld in Xamarin
- Wireframes voor design van app op elk platform

Fase 5: stage - uitwerking

- Mobiele toepassing
- Documentatie in code
- Klassendiagram



Sam Debruyn

19/11/2015

9

Sam Debruyn

19/11/2015

30

od'icee



Sam Debruyn

19/11/2015

ASSOCIATIE
KU LEUVEN

31

9.3.3 Presentatie 3: slotpresentatie

Inhoud

- Inleiding
- Omschrijving van de klant
- Projectomschrijving
- Aanpak
- Producten
- Conclusie
- Zelfreflectie

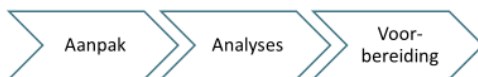
Cross-platform mobile apps met Xamarin 15/01/2016 2

Inleiding

Van probleem naar oplossing

2 delen:

- 1) bachelorproef: voorbereiding (*plan-prepare-design*)
- 2) stage: implementatie (*implement-test*)



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Wie is de klant?

Opdrachtgever: RealDolmen

Klant: RealDolmen Education



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

RealDolmen Education

Instructor-Led Training (ILT)

Klassieke opleiding met docent

E-Learning Projects

Digitale leeromgeving, student ondersteund door software

Technical Writing

Documentatie van software

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Het project: situatie vandaag



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Het project: vraag van de klant

- Alternatief voor website op smartphones & tablets
- Zichtbaarheid in applicatiewinkels
- ✓ Eén mobiele app, meerdere platformen
- ? In theorie onmogelijk



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Aanpak: 4 fases



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

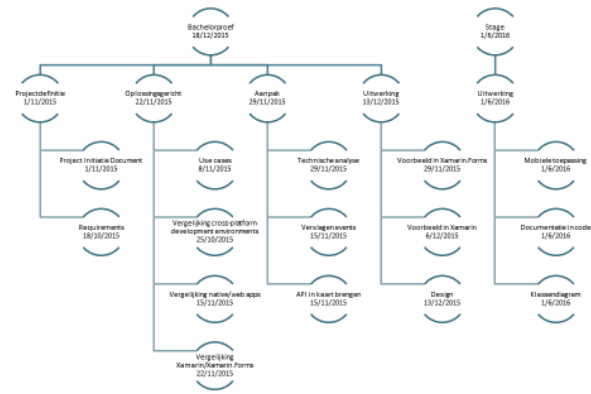
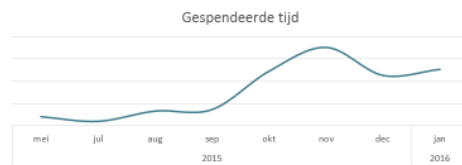
Aanpak: projectmatig

Tijdsbesteding

Contacthistoriek

Project Planning

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie



Cross-platform mobile apps met Xamarin

15/01/2016

9

Cross-platform mobile apps met Xamarin

15/01/2016

10

Fase 1: projectdefinitie

Project Initiatie Document (PID)

Contract
Risico's
Scope

Requirements analyse

Mobiele platformen & versies
Platformspecifieke user experience

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Cross-platform mobile apps met Xamarin

15/01/2016

11

Fase 1: projectdefinitie

API documentatie

Ook fase 4

Use cases

Beschrijvingen
Diagram

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Cross-platform mobile apps met Xamarin

15/01/2016

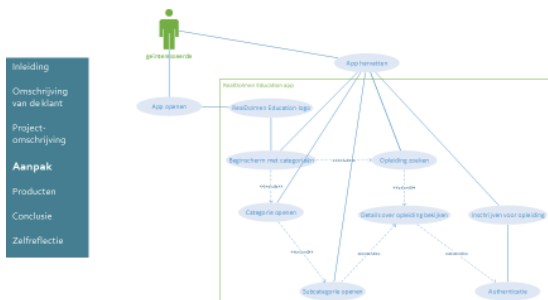
12

Cross-platform mobile apps met Xamarin

15/01/2016

13

Fase 1: projectdefinitie



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Cross-platform mobile apps met Xamarin

15/01/2016

13

Fase 2: oplossingen zoeken

Cross-platform development environment (IDE)

Shared code

Vergelijking cross-platform IDE's

Eigenschappen gescoord
→ twee soorten apps

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Cross-platform mobile apps met Xamarin

15/01/2016

14

Cross-platform mobile apps met Xamarin

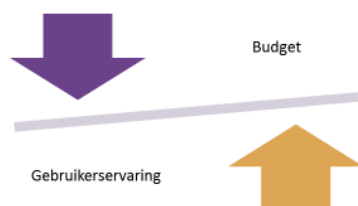
15/01/2016

14

Fase 2: oplossingen zoeken

Vergelijking native & web apps

Onderliggende architectuur (CPU)



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Cross-platform mobile apps met Xamarin

15/01/2016

15

Fase 3: oplossingen vergelijken

Resultaat uit vorige fases: Xamarin Inc.

Vergelijking Xamarin & Xamarin.Forms



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Cross-platform mobile apps met Xamarin

15/01/2016

16

Cross-platform mobile apps met Xamarin

15/01/2016

16

Fase 3: oplossingen vergelijken

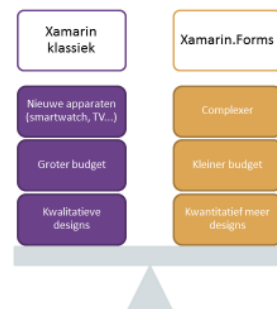
Demo apps

Ook fase 4

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Fase 3: oplossingen vergelijken

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie



Fase 4: implementatie voorbereiden

Technische analyse

NuGet packages
Standaarden (OAuth voor authenticatie, REST voor communicatie met API...)
Architectuur

Wireframes

Evenementen

AZUG
MADN

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Analyses

Vergelijking cross-platform development environments ②

Vergelijking native & web apps ②③

Vergelijking Xamarin & Xamarin.Forms ③

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Demo apps

<https://dataservice.kuleuven.be/v1/program2015/search.json?q=programId:s1854577>

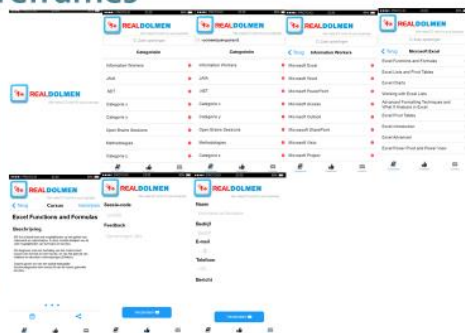
Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

```

{
  "Program": "Computer Science",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to computer science, covering topics such as algorithms, data structures, and programming."
},
{
  "Program": "Mathematics",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to mathematics, covering topics such as calculus, linear algebra, and probability."
},
{
  "Program": "Physics",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to physics, covering topics such as mechanics, electromagnetism, and quantum mechanics."
},
{
  "Program": "Chemistry",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to chemistry, covering topics such as atomic structure, chemical reactions, and thermodynamics."
},
{
  "Program": "Biology",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to biology, covering topics such as cell biology, genetics, and evolution."
},
{
  "Program": "Psychology",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to psychology, covering topics such as cognitive psychology, social psychology, and developmental psychology."
},
{
  "Program": "History",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to history, covering topics such as world history, European history, and American history."
},
{
  "Program": "Literature",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to literature, covering topics such as English literature, world literature, and comparative literature."
},
{
  "Program": "Art",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to art, covering topics such as visual arts, music, and theater."
},
{
  "Program": "Sports",
  "Year": 2015,
  "Credits": 10,
  "Description": "Introduction to sports, covering topics such as sports science, sports management, and sports law."
}

```

Wireframes



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

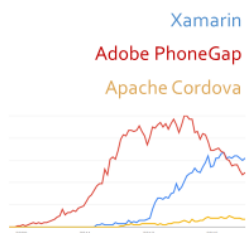
Conclusie

Project klaar voor implementatie

Xamarin app + deelp producten

Jonge trend die sterk evolueert

Grote vraag, aanbod moet nog volgen
Ervaring voor ontwikkelaar niet optimaal



Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Conclusie

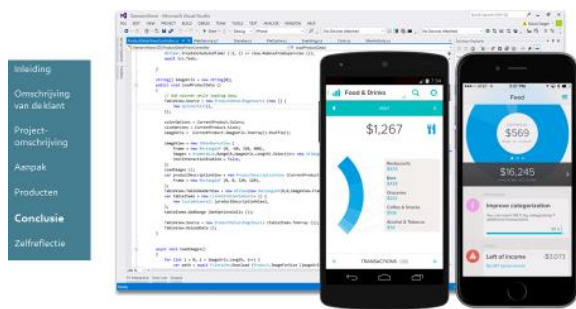
Stakeholders:

Mogelijke keuzes:

1. Elkaars mogelijkheden overnemen
2. Ondersteuning voor cross-platform development
3. Eigen platform afschermen om unieke mogelijkheden te beschermen

Inleiding
Omschrijving van de klant
Project-omschrijving
Aanpak
Producten
Conclusie
Zelfreflectie

Conclusie



Zelfreflectie

Bewust gekozen: interessante trend

Waardevolle ervaring

Interessant voor diepgaand onderzoek

Developer → technical specialist (+ project manager)

Voorbereiding afgewerkt zoals gepland



Vragen?



9.4 Verslagen events

9.4.1 Verslag AZUG: 3 sessions and a BBQ

22 augustus 2015, RealDolmen, Huizingen

9.4.1.1 *Wie of wat is AZUG?*

AZUG (Azure User Group) is een groep Belgische softwareontwikkelaars die veel werken met de diensten die Microsoft aanbiedt onder de noemer Microsoft Azure.

Ze worden gesponsord door grote IT-bedrijven actief op de Belgische markt die zich focussen op software-ontwikkeling met het .NET platform: Microsoft, RealDolmen, Ordina, Cegeka, Cronos, JetBrains, Capgemini...

Met deze sponsoring organiseren ze evenementen waarop sessies gehouden worden die nieuwe mogelijkheden van Azure uit de doeken doen.

9.4.1.2 *Agenda & sprekers*

- 8h30: ontbijt
- 9h: sessie Azure Service Fabric, next generation Platform as a Service
Spreker: Yves Goeleven (oprichter MessageHandler.net, Windows Azure Most Valuable Professional)
- 10h15: sessie Authenticating Mobile Apps through Windows Azure Active Directory
Spreker: Kurt Claeys (Microsoft Certified Trainer, Azure Certified Technology Specialist, .NET Most Valuable Professional, Xamarin Certified Mobile Developer)
- 11h30: sessie Get more than a cache back! The Microsoft Azure (Redis) Cache
Spreker: Maarten Balliauw (Software Engineer bij Microsoft & NuGet, oprichter van MyGet.org)
- 12h30: BBQ & networking

9.4.1.3 *Verslag*

9.4.1.3.1 *Verworven kennis*

Voor dit evenement wist ik wel ongeveer wat de mogelijkheden van Azure zijn en hoe uitgebreid dit platform is, maar was mijn kennis ervan nog niet zo diep als erna.

Service Fabric is aardig op weg om de opvolger te worden van de bestaande Cloud Services en App Services, maar zit ook nog wel in een vroeg stadium.

Redis is de snelste manier van caching, zowel op een server, in de cloud als lokaal. Het knappe aan Redis blijft de ongelooflijke eenvoudigheid en snelheid, ook tijdens de demo werd dit nog maar eens duidelijk.

Het interessantste deel voor mijn bachelorproef was het tweede luik. Hierin gaf Kurt Claeys een introductie tot het gebruik van Azure Active Directory als authenticatiemodule bij een ASP.NET web API in combinatie met een alle soorten apps. Ook Xamarin kwam aan bod.

9.4.1.3.2 *Persoonlijke mening en ervaringen*

Dit is het eerste evenement van AZUG dat ik zelf bijgewoond heb en ik beklagde me meteen dat ik niet eerder lid geworden ben van deze groep sympathieke ontwikkelaars. Je leert enorm veel bij op hun evenementen en de toepassingen gaan breder dan Microsoft Azure zelf, ook al is het zo'n uitgebreid platform.

9.4.1.3.3 TI-competenties

De volgende competenties en kerndoelen kwamen aan bod:

- 7. Op een adequate wijze IT-oplossingen configureren, beveiligen en aanpassen zodat deze beantwoorden aan de veranderende behoeften van de organisatie.
 - 7.3 Toepassingen, databanken en systemen als gevolg van wijzigende omstandigheden aanpassen.
- 9. IT-taken kwaliteitsvol uitvoeren zodat het resultaat voldoet aan de eisen van een steeds wisselende omgeving.
 - 9.1 Zelfstandig nieuwe IT-concepten en principes leren hanteren.

9.4.2 Verslag AZUG: AzureCon

29 september 2015, Microsoft, Zaventem

9.4.2.1 *Wie of wat is AZUG?*

AZUG (Azure User Group) is een groep Belgische softwareontwikkelaars die veel werken met de diensten die Microsoft aanbiedt onder de noemer Microsoft Azure.

Ze worden gesponsord door grote IT-bedrijven actief op de Belgische markt die zich focussen op software-ontwikkeling met het .NET platform: Microsoft, RealDolmen, Ordina, Cegeka, Cronos, JetBrains, Capgemini...

Met deze sponsoring organiseren ze evenementen waarop sessies gehouden worden die nieuwe mogelijkheden van Azure uit de doeken doen.

9.4.2.2 *Agenda & sprekers*

AzureCon is een jaarlijkse conferentie die wereldwijd wordt gelivestreamd zodat geïnteresseerden deze zowel van thuis uit als in een lokaal kantoor van Microsoft kunnen volgen. AZUG organiseerde een livestream op het hoofdkantoor van Microsoft Belux in Zaventem.

- 17h30: ontvangst
- 18h: keynote livestream
 - Journey to the intelligent cloud
Spreker: Scott Guthrie (Executive Vice President Cloud & Enterprise)
 - Building apps with the intelligent cloud
Spreker: Bill Staples (Corporate Vice President Azure App Platform)
 - Transforming data into intelligent action
Spreker: T.K. "Ranga" Rengarajan (Corporate Vice President Azure Data Platform)
 - Infrastructure for the intelligent cloud
Spreker: Jason Zander (Corporate Vice President Azure Development)
 - Slotwoord
Spreker: Satya Nadella (CEO)
- 21h30: networking

9.4.2.3 *Verslag*

9.4.2.3.1 *Verworven kennis*

Microsoft Azure is een gigantisch platform dat elke maand nog nieuwe functies krijgt. De kracht van Azure zit hem in de schaalbaarheid. Het is ongelooflijk eenvoudig om snel duizenden servers op te starten om je website op te draaien zodat deze niet offline gaat bij een plotse bezoekerstoename.

Tijdens AzureConf werden nieuwe functies toegelicht en gaven klanten voorbeelden van hoe ze Azure gebruiken om hun bedrijf vooruit te helpen. Centraal stond het Internet of Things. Autobedrijven lieten zien hoe de gegevens van al hun sensoren in realtime naar Azure gestreamd worden. Vervolgens werd gedemonstreerd hoe Azure gebruikt kan worden om een Data Lake op te zetten met al deze gegevens en deze daarna te doorploegen met de nieuwe hybride programmeertaal USQL.

In de context van mijn bachelorproef sprongen enerzijds de eenvoudige tools om een web API te bouwen in het oog en anderzijds de mogelijkheden voor continuous integration. Het wordt heel eenvoudig om mobiele toepassingen automatisch te laten compileren en signeren in de cloud.

9.4.2.3.2 Persoonlijke meningen & ervaringen

Microsoft toonde nog maar eens aan hoe groot ze wel niet zijn en waartoe ze allemaal in staat zijn. De focus ligt al lang niet meer op de verkoop van Windows, XBox en andere oudere producten. De toekomst zit duidelijk in de cloud en Azure speelt hier perfect op in. Ze ondersteunen ook meer en meer open technologieën waardoor Azure meer flexibiliteit krijgt.

Het was een heel interessant evenement en werd uitstekend verzorgd door Microsoft België en AZUG. Ook rond het project van mijn bachelorproef heb ik veel nuttige kennis opgedaan.

9.4.2.3.3 TI-competenties

De volgende competenties en kerndoelen kwamen aan bod:

- 7. Op een adequate wijze IT-oplossingen configureren, beveiligen en aanpassen zodat deze beantwoorden aan de veranderende behoeften van de organisatie.
 - 7.3 Toepassingen, databanken en systemen als gevolg van wijzigende omstandigheden aanpassen.
- 9. IT-taken kwaliteitsvol uitvoeren zodat het resultaat voldoet aan de eisen van een steeds wisselende omgeving.
 - 9.1 Zelfstandig nieuwe IT-concepten en principes leren hanteren.

9.4.3 Verslag MADN: Securing and consuming a Web API with OAuth

15 december 2015, RealDolmen, Kontich

9.4.3.1 Wie of wat is MADN?

MADN staat voor Metro App Developer Network. Toen Microsoft Windows 8 lanceerde, lanceerden ze ook meteen een volledig nieuwe manier om apps voor Windows te maken. De klassieke Windows Forms en WPF apps waren uit den boze en de nieuwe stijl van Windows apps heette Metro. De naam veranderde later naar Modern UI, maar dat hield enkele Belgische softwareontwikkelaars niet tegen om zich te verenigen in een nieuwe user group, gelijkaardig aan AZUG en VISUG.

Ze organiseren maandelijks evenementen waar sessies en dergelijke gegeven worden om ontwikkelaars te helpen betere apps te maken en de laatste nieuwe methodieken uit de doeken te doen.

9.4.3.2 Agenda & sprekers

- 18h: ontvangst
- 18h30: *Securing and consuming a Web API with OAuth* door Wesley Cabus (Lead .NET developer, RealDolmen)
- 20h30: networking

9.4.3.3 Verslag

9.4.3.3.1 Verworven kennis

OAuth is een complex mechanisme met verschillende mogelijke implementaties om webtoepassingen te beveiligen. Het is ontworpen om op een gestandaardiseerde manier gebruikers te authenticeren en te autoriseren bij een webserver. Vervolgens moeten de apps die gebruik maken van de diensten van die webserver dan op een gelijkaardige manier gebouwd zijn zodat ze correct kunnen omgaan met het authenticatiemechanisme op de webserver.

Bij OAuth gaat het om de uitwisseling van gebruikersgegevens aan de hand van unieke codes met beperkte geldigheid. Er zijn verschillende 'flows' die beschrijven hoe de overdracht van die codes moet verlopen. Het is aan de softwareontwikkelaar om de juiste flow te kiezen voor zijn/haar toepassing. Dit hangt af van de soort toepassing die gebruik maakt van de diensten van de webserver.

9.4.3.3.2 Persoonlijke mening & ervaringen

Het is geen sinecure om de juiste 'flow' te vinden voor je webtoepassing en deze daarna correct in te stellen, zowel aan de kant van de webserver als aan de kant van de app. Ook bij de app die ik tijdens mijn stage zal ontwikkelen, zal ik hier gebruik van moeten maken.

Wesley heeft hier duidelijk ervaring mee en kent de valkuilen waar ontwikkelaars in lopen. Het was een heel leerrijke sessie en met de verworven kennis zal ik mezelf sowieso tijd besparen tijdens mijn stage.

9.4.3.3.3 TI-competenties

De volgende competenties en kerndoelen kwamen aan bod:

- 5. Gegevens verzamelen, opslaan en ter beschikking stellen zodat deze op een correcte en gebruiksvriendelijke manier kunnen worden opgevraagd

- 5.5 De gegevens in een aangepast formaat presenteren en rekening houden met de bevoegdheden
- 7. Op een adequate wijze IT-oplossingen configureren, beveiligen en aanpassen zodat deze beantwoorden aan de veranderende behoeften van de organisatie.
 - 7.2 Systemen, toepassingen of databanken beveiligen en voorstellen doen om beveiligingsrichtlijnen en -procedures aan te passen
- 9. IT-taken kwaliteitsvol uitvoeren zodat het resultaat voldoet aan de eisen van een steeds wisselende omgeving.
 - 9.1 Zelfstandig nieuwe IT- concepten en principes leren hanteren
 - 9.3 Een gebruiksvriendelijke IT-oplossing uitwerken die aanpasbaar is aan de veranderende behoefte

9.5 Project Initiatie Document

9.5.1 Projectfiche

DocumentID	BP1516-SDRDEM-PID
ProjectID	BP1516-SDRDEM
Projectnaam	Bachelorproef – RealDolmen Education mobile apps
Datum	18/12/2015
Auteur	Samuel Debruyne
Owner	RealDolmen
Klant	RealDolmen Education

9.5.2 Updatehistoriek

Datum	Omschrijving van wijzigingen
25/10/2015	Eerste versie, structuur PID en aanvullen projectdefinitie
31/10/2015	Verwerking feedback Gert Cauwenbergh en Nico Vermeir
17/11/2015	Verwerking feedback Serge van Cleynenbreugel en Gert Cauwenbergh
28/11/2015	Update risico's met grotere kans op problemen met OS X-toestel
4/12/2015	Update configuratiebeheer broncode
11/12/2015	Spelling en grammatica

9.5.3 Distributielijst

Naam	Functie	Instelling
Samuel Debruyne	Eindverantwoordelijke project	Odisee
Yvan Rooseleer	Opleidingshoofd	Odisee
Chantal Teerlinck	Begeleider bachelorproef	Odisee
Hans Vandenbogaerde	Begeleider bachelorproef	Odisee
Gert Cauwenbergh	Begeleider bachelorproef	Odisee
Serge Van Cleynenbreugel	Projectbegeleider	Odisee
Nico Vermeir	Projectbegeleider	RealDolmen

9.5.4 Projectdefinitie

9.5.4.1 *Achtergrond*

RealDolmen is een Belgisch bedrijf gespecialiseerd in de integratie van bestaande en eigen ICT-producten. Het bedrijf is ontstaan in 2008 na de fusie van Real Software en Dolmen en stelt vandaag meer dan 1500 mensen te werk. RealDolmen heeft verschillende afdelingen die elk gespecialiseerd zijn in een bepaald technisch domein, zoals bvb. .NET software, Java software, netwerken en systemen, business intelligence...

RealDolmen biedt onder het merk RealDolmen Education verschillende opleidingen aan in het Nederlands, het Frans en het Engels. De opleidingen worden gegeven door medewerkers van RealDolmen en zijn bedoeld voor geïnteresseerden uit andere bedrijven en in sommige gevallen ook voor eigen personeel.

Naast instructor-led training (ILT), biedt het bedrijf ook andere diensten aan zoals ondersteuning bij het schrijven van documentatie, certificatie of het opzetten van een e-learning omgeving.

Momenteel kan het aanbod bekeken worden via een website. Hier kan gezocht worden naar opleidingen, maar ze staan zelf ook ingedeeld in categorieën. Deze website is niet handig voor gebruik op smartphones.

RealDolmen Education zou hun aanbod aan opleidingen graag in een mobiele applicatie gieten. De afdeling zou graag één mobiele applicatie hebben die werkt op de drie grootste mobiele platformen (Android, iOS en Windows Phone). Zo spaart men tijd en kosten uit voor de ontwikkeling ervan en kunnen er toch heel wat potentiële klanten meer bereikt worden.

9.5.4.2 Objectieven

De requirements worden beschreven in 9.6 REQUIREMENTS ANALYSE.

- RealDolmen Education heeft een app die ze kunnen publiceren in de app stores van Apple, Google en Microsoft.
- De applicatie bereikt zoveel mogelijk mensen door beschikbaar te zijn
 - in 3 talen: Engels, Nederlands, Frans
 - op 3 platformen: Android, iOS, Windows Phone
- In de applicatie staan de opleidingen ingedeeld op een manier die sneller te navigeren is op een smartphone dan via de website.

9.5.4.3 *Gewenste resultaat*

RealDolmen Education wil een vaste plek hebben op de smartphones van (potentiële) klanten. Hiermee poogt men het volgende te bereiken:

- Meer mensen bekijken het aanbod aan opleidingen.
- Meer mensen raken geïnteresseerd in opleidingen die RealDolmen Education aanbiedt.
- Geïnteresseerden vinden makkelijker de opleiding waar ze naar op zoek zijn.
- Het aanbod aan opleidingen kan geraadpleegd worden zonder gebruik te moeten maken van een tablet of computer en een webbrowser.

9.5.4.4 Reikwijdte (scope)

9.5.4.4.1 Binnen de scope

- 1) De ontwikkeling van een applicatie voor RealDolmen Education die werkt op meerdere platformen en die voldoet aan de eerder vermelde (niet-)functionele eisen.
- 2) Het testen van de applicatie op de drie verschillende platformen. Het is voldoende om ze in een simulator van het desbetreffende platform te testen.
- 3) De ontwikkeling van de visuele weergave van de applicatie. Met uitzondering van de RealDolmen huisstijl, valt het design van de applicatie binnen het project.
- 4) De vertaling van het design naar de drie talen. De gegevens die getoond moeten worden, zijn reeds beschikbaar in drie talen, maar het design van de applicatie is dat niet. Hier moet de ontwikkelaar voor zorgen.

9.5.4.4.2 Buiten de scope

- 1) Er werd voor een stageproject in academiejaar 2014-2015 een API⁷ ontwikkeld die alle gegevens aanbiedt die nodig zijn voor deze applicatie. Het is aangewezen om deze API te gebruiken. De ontwikkeling van een eigen API ligt buiten de scope van dit project.
- 2) De publicatie van de applicatie op een manier dat eindgebruikers deze kunnen installeren. Indien de applicatie goedgekeurd wordt, zal RealDolmen hier zelf voor zorgen.
- 3) RealDolmen levert een huisstijl aan die gebruikt moet worden voor het design van de applicatie.
- 4) RealDolmen levert de nodige hardware en software aan die nodig is voor de ontwikkeling van de applicatie en al zijn onderdelen.

⁷ API: Application Programming Interface. Dit is een systeem dat gegevens beschikbaar stelt in een formaat dat door softwareontwikkelaars gebruikt kan

worden om weer te geven in een applicatie of om te gebruiken bij de verwerking van dezelfde gegevens.

9.5.4.5 *Beperkingen en assumpties*

9.5.4.5.1 Beperkingen

RealDolmen heeft ervoor gekozen om te werken met het .NET platform. Dit betekent dat de mobiele toepassing in de programmeertaal C# geschreven moet worden.

Het project moet uitgewerkt worden met Xamarin (geen Xamarin.Forms). Doch is het interessant om tijdens de analysefase de alternatieven eens te bekijken.

9.5.4.5.2 Assumpties

- De gegevens die nodig zijn als invoerbron van de applicatie zijn reeds beschikbaar in een web API. Deze API blijft werken gedurende de periode van ontwikkeling van de toepassing.
- De huisstijl van RealDolmen wijzigt niet tijdens de ontwikkeling van de toepassing.
- Testen mogen gebeuren in emulators en er wordt vanuit gegaan dat deze emulators een realistisch beeld geven van de toestellen die ze simuleren. Een app die werkt in een emulator van een mobiel platform, werkt ook op een toestel dat draait op het desbetreffende mobiele platform.
- De projectbegeleider van RealDolmen geeft de requirements van de mobiele toepassing duidelijk weer in naam van de klant (RealDolmen Education).

9.5.4.5.3 Afhankelijkheden

- Het visuele design van de applicatie is afhankelijk van de huisstijl van RealDolmen en tijdige aflevering hiervan.
- De applicatie kan enkel werken indien de aangereikte web API operationeel is.
- Aangezien de applicatie op meerdere platformen moet werken, is men sterk afhankelijk van de software waarmee de applicatie ontwikkeld wordt. Deze software is verantwoordelijk voor het omzetten van generieke code naar code die werkt op een specifiek mobiel platform.

9.5.4.6 *Betrokken organisaties (stakeholders)*

- Odisee UC: Samuel Debruyn ontwikkelt als stageopdracht bij Odisee UC de applicatie voor dit project en voert eveneens de voorafgaande analyses uit. Bij Odisee UC zijn Yvan Rooseleer, Chantal Teerlinck, Hans Vandenbogaerde, Gert Cauwenbergh en Serge Van Cleynenbreugel verantwoordelijk voor de begeleiding hiervan.
- RealDolmen: Nico Vermeir begeleidt dit project vanuit RealDolmen.
- RealDolmen Education: de klant.

9.5.4.7 *Gerelateerde projecten*

- In academiejaar 2014-2015 heeft Jens Van Den Driessche als bachelorproef een RESTful web API ontwikkeld in ASP.NET dat gebruikt dient te worden als invoerbron voor de applicatie.
- Sam Van Cutsem ontwikkelde in academiejaar 2014-2015 een gelijkaardige applicatie die niet volledig voldeed aan de wensen van de klant.

9.5.5 Aanpak

Er zijn verschillende mogelijke manieren om apps te creëren voor meerdere platformen. Deze zullen geëvalueerd en onderling vergeleken worden in de analysefase (oktober 2015 – december 2015). RealDolmen stelt echter zelf een traject voor dat gevolgd moet worden om de applicatie te ontwikkelen.

9.5.5.1 Uitgewerkte analyses

In de analysefase worden er enkele mogelijke ontwikkelomgevingen vergeleken. Hierbij worden verschillende aspecten nauwkeurig onderzocht en geëvalueerd. De ontwikkelomgevingen die vergeleken worden zijn sowieso al in staat om de wensen van de klant te vervullen. De aspecten die vergeleken worden zijn de volgende:

- Efficiëntie van de ontwikkeling
- De manier waarop de code gedeeld wordt tussen de mobiele platformen waarvoor ontwikkeld wordt
- De hulpmiddelen die gebruikt kunnen worden tijdens het creëren van het visuele design
- Ondersteuning voor emulators bij het testen

9.5.5.1.1 Xamarin

De ontwikkelomgeving Xamarin kan gebruikt worden om een toepassing te ontwikkelen in de programmeertaal C# met behulp van het .NET Framework. Deze 'generische' code wordt nadien omgezet naar code die native⁸ werkt op Android, iOS en Windows.

Bij Xamarin wordt zoveel mogelijk achterliggende code gedeeld over mobiele platformen heen, maar wordt het visuele design apart ontwikkeld voor elk platform opdat dit optimaal zou werken op elk specifiek mobiel platform.

9.5.5.1.2 Xamarin.Forms

Xamarin.Forms is gerelateerd aan Xamarin, maar is op zich wel een ander product. Hierbij wordt de hoeveelheid gedeelde code veel groter aangezien ook het visuele design door alle mobiele platformen gedeeld wordt.

9.5.5.2 Gekozen oplossing

De oplossing die voorgesteld werd door RealDolmen en gevolgd dient te worden is een mobiele applicatie ontwikkeld met Xamarin.

Voor de ontwikkeling zelf zal Microsoft Visual Studio met Xamarin gebruikt worden en tests kunnen uitgevoerd worden via emulators. Om iOS te emuleren is een toestel met OS X vereist.

9.5.5.3 Project Plan

Zie bijlage 9.8 PROJECT PLANNING.

⁸ Het begrip *native code* wordt verduidelijkt in 9.12 VERGELIJKING NATIVE EN WEB APPS.

9.5.5.4 Risico's

Het gebruik van Xamarin houdt verschillende risico's in.

Risico	Kans (/10)	Impact (/10)	Score	Maatregel
De ontwikkeling van aparte visuele designs voor elk mobiel platform kan tot onvoorziene omstandigheden leiden.	7	7	49	Voorafgaande studie doen over de ontwikkeling van het visuele design voor elk mobiel platform.
De gedeelde code kan verschillende resultaten geven op verschillende mobiele platformen.	2	4	8	Debugging tools zoeken om voor elk mobiel platform apart te debuggen.
Voor ondersteuning is men volledig afhankelijk van de Xamaringsgemeenschap.	10	1	10	Licentie aankopen om extra ondersteuning te krijgen.
Om de applicatie op iOS te kunnen testen is een toestel met OS X vereist. Aangezien het toestel verder weinig gebruikt wordt, kunnen er problemen opduiken op dit toestel zelf.	7	9	63	Een ander toestel met OS X zoeken of tijd voorzien om configuratieproblemen op te lossen.

Na een analyse van de risico's wordt besloten om voorafgaand aan de ontwikkeling reeds enkele online lessen te volgen over de ontwikkeling voor de verschillende mobiele platformen.

Er wordt eveneens voldoende tijd ingeschat om de configuratie van het OS X-toestel in te stellen.

9.5.5.5 Samenwerking en controle

De ontwikkeling wordt wekelijks geëvalueerd door de projectbegeleider bij RealDolmen. Er worden eveneens tussentijdse evaluaties voorzien door de projectbegeleiders bij Odisee. Het project loopt 60 dagen en dient volledig af te zijn aan het einde van deze periode.

9.5.6 Beheersaspecten

9.5.6.1 *Kwaliteitscontrole*

Ter kwaliteitscontrole wordt er begeleiding georganiseerd vanuit Odisee en vanuit RealDolmen. Elke vrijdag is een halve dag begeleiding voorzien met Nico Vermeir. Daarnaast voorziet ook Odisee feedback, zowel via e-mail als tijdens afspraken met de projectbegeleider Serge Van Cleynenbreugel.

Voor de ontwikkeling van de applicatie worden in de sector erkende methodologieën systematisch toegepast. Dit moet bijdragen tot een constante kwaliteit, vereenvoudigde manier van werken en een eenvoudige evaluatie.

9.5.6.2 *Configuratiebeheer*

Het configuratiebeheer werd reeds opgesteld door de projectbegeleiders bij Odisee. Zie 'Instructie Configuration Management'.

De documenten worden geëxporteerd naar PDF, de broncode wordt ingepakt in een gecomprimeerde map. De naam van het bestand is telkens de titel van de deliverable. Bij meerdere versies wordt na de titel de datum vermeld om ingewikkeld versiebeheer te vermijden.

Broncode voor de analysefase wordt beschikbaar gemaakt via GitHub. Er staat een document met links naar de GitHub repositories in de gedeelde map op Google Drive.

Tijdens de uitwerking zelf, de stage, wordt de code beschikbaar gemaakt via de Team Foundation Server van RealDolmen.

9.6 Requirements analyse

9.6.1 Functionele requirements

- 1) De applicatie werkt op smartphones op basis van Google Android 4.4 of nieuwer.
- 2) De applicatie werkt op smartphones op basis van Apple iOS (iPhone) 8 en 9.
- 3) De applicatie werkt op smartphones op basis van Microsoft Windows Phone (Windows Mobile) 8.1 en 10.
- 4) Alle opleidingen moeten individueel weergegeven kunnen worden met al hun informatie zoals beschikbaar op de website.
- 5) De opleidingen moet bekeken en gefilterd kunnen worden op categorie en op subcategorie.
- 6) Er moet gezocht kunnen worden naar opleidingen via de applicatie.
- 7) Gebruikers moeten zich kunnen inschrijven voor een opleiding (hiervoor is authenticatie nodig).
- 8) De website wordt aangeboden in het Engels, in het Nederlands en in het Frans. De applicatie moet beschikbaar zijn in dezelfde drie talen. Er mogen geen onvertaalde delen zichtbaar zijn.
- 9) Er moet caching voorzien worden zodat de gegevens niet elke keer opnieuw opgehaald moeten worden. Gegevens die reeds opgehaald zijn, zijn dus offline beschikbaar.
- 10) Een gast op een les moet feedback over de les kunnen doorsturen.

9.6.2 Niet-functionele requirements

- 1) De huisstijl van RealDolmen moet duidelijk zichtbaar zijn in de applicatie. Deze bestaat uit enkele kleuren (rood en blauw), logo's en lettertypes.
- 2) De applicatie moet op elk platform op een logische manier voor de gebruiker werken (zie Platform-specifieke user experience).
- 3) De applicatie moet vlot en snel werken.

9.6.2.1 *Platformspecifieke user experience*

Elk mobiel platform (Android, iOS, Windows Phone...) gaat op een andere manier om met vaak terugkerende patronen. Zo hebben Android en Windows Phone een *Terug*-knop onderaan op het toestel zelf, terwijl dezelfde knop bij iOS meestal links bovenaan op een scherm staat. Zo zijn er talrijke patronen die eigen zijn aan een platform.

Het respecteren van deze patronen zorgt ervoor dat de gebruiker veel minder moet nadenken over het gebruik van de applicatie en alles steeds op dezelfde plaats terugvindt. Het is zelden interessant om van deze patronen af te wijken gezien de bijkomende complexiteit.

9.7 Lijst deliverables

9.7.1 Bachelorproef

9.7.1.1 *Persoonlijke deliverables*

- Curriculum Vitae
- Motivatiebrief Delaware Consulting
- Motivatiebrief Euricom
- Motivatiebrief RealDolmen – big data
- Motivatiebrief RealDolmen – .NET
- Time tracker
Time tracking gebeurt via de app toggl.com. Er kunnen rapporten gegenereerd worden in PDF of in CSV (Excel).
- Project Planning
- Contacthistoriek
- Bibliografie

9.7.1.2 *Projectspecifieke deliverables*

9.7.1.2.1 Projectomschrijving

- Project Initiatie Document
- Functionele en niet-functionele requirements
- Technische analyse
- Voorbeelden van use cases

9.7.1.2.2 Vergelijkende analyses

- Vergelijking van mogelijkheden en eigenschappen van cross-platform development environments
- Vergelijking van prestaties en uitwerking bij native apps en web apps
- Vergelijking tussen Xamarin en Xamarin.Forms

9.7.1.2.3 Uitwerkingen

- Uitwerking van voorbeeld met Xamarin.Forms
- Uitwerking van voorbeeld met Xamarin (klassiek)
 - Zaterdag 22/08: sessies bij AZUG: verslag
 - Dinsdag 29/03: AzureCon met AZUG: verslag
 - Dinsdag 15/12: sessie OAuth bij MADN: verslag
- Wireframes voor design van app op Android, iOS en Windows Phone

- In kaart brengen van bestaande web API

9.7.2 Stage

9.7.2.1 *Persoonlijke deliverables*

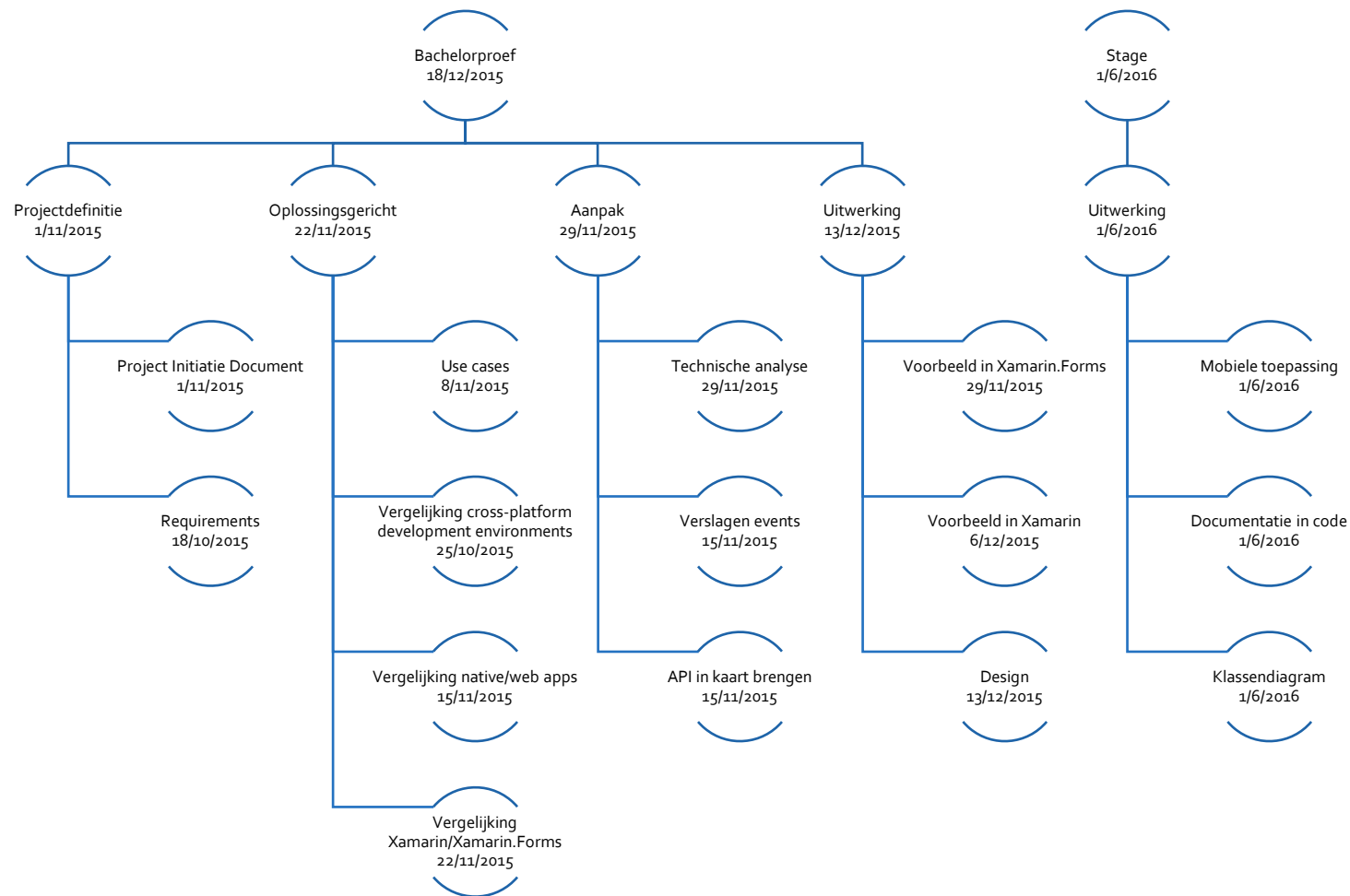
- Time tracker (vervolg bachelorproef)
- Project Planning (zie bachelorproef)
- Contacthistoriek (vervolg bachelorproef)
- Bibliografie

9.7.2.2 *Projectspecifieke deliverables*

9.7.2.2.1 Uitwerking

- De applicatie voor de 3 platformen
- Documentatie in de code
- Klassendiagram

9.8 Project planning



9.9 Use cases

9.9.1 Beschrijvingen

Optionele stappen worden cursief vermeld in een scenario.

9.9.1.1 Actoren

- Geïnteresseerde
- Klant

Een *geïnteresseerde* kan *klant* worden door zich te authenticeren met een gebruikersnaam en wachtwoord.

9.9.1.2 Een geïnteresseerde zoekt naar een opleiding

Primaire actor: geïnteresseerde

Secundaire actoren: geen

Preconditie: De geïnteresseerde heeft de RealDolmen Education app geïnstalleerd op zijn smartphone.

Postconditie: De geïnteresseerde heeft de volgende informatie over de opleiding:

- Duur (aantal dagen)
- Beschrijving
- Inhoud van de opleiding
- Doelgroep
- Doel van de opleiding
- Vereiste voorkennis
- Methode waarop de opleiding gegeven wordt

9.9.1.2.1 Hoofdscenario

- 1) De geïnteresseerde opent de RealDolmen Education app op zijn smartphone.
- 2) De geïnteresseerde wacht totdat de RealDolmen Education app geladen is, maar ziet ondertussen al het logo van RealDolmen (Education).
- 3) De geïnteresseerde tikt op de zoekbalk op het scherm.
 - a. Indien de geïnteresseerde een zoekknop heeft op zijn/haar apparaat, kan deze als alternatief gebruikt worden.
- 4) De geïnteresseerde krijgt een toetsenbord te zien.
- 5) De geïnteresseerde typt de naam van de opleiding in waarnaar hij/zij op zoek is.
- 6) De geïnteresseerde drukt op de zoekknop of op *enter*.
- 7) *De geïnteresseerde krijgt de resultaten van de zoekopdracht te zien en ziet de opleiding waar hij of zij naar op zoek is in de lijst.*
- 8) *De geïnteresseerde tikt vervolgens op die opleiding.*
 - a. Indien er slechts één opleiding gevonden is, wordt deze meteen getoond en gaat men van stap 6 meteen naar stap 9.
- 9) De geïnteresseerde krijgt de details over zijn opleiding te zien.
- 10) *Mogelijks moet er nog gescrold worden door met de vinger over het scherm te swipen.*

9.9.1.2.2 Uitbreidingen

Dit scenario kan verder lopen in use case Een klant schrijft zich in voor een opleiding.

9.9.1.3 Een klant schrijft zich in voor een opleiding

Primaire actor: klant

Secundaire actoren: geen

Preconditie: De klant heeft de RealDolmen Education app geïnstalleerd op zijn

smartphone. De klant heeft de details van de opleiding reeds open staan in de app.

Postconditie: De klant is ingeschreven voor de opleiding.

9.9.1.3.1 Hoofdscenario

- 1) De klant klikt op de Inschrijfknoop.
- 2) De klant ziet een scherm waar een gebruikersnaam en wachtwoord ingevoerd kunnen worden. Er komt ook een toetsenbord tevoorschijn.
- 3) De klant geeft zijn gebruikersnaam en wachtwoord in en drukt op *enter* of op een knop om zich aan te melden.
 - a. De ingevoerde gegevens waren juist. De klant gaat verder met stap 4.
 - b. De ingevoerde gegevens waren fout. De klant krijgt een foutmelding te zien en kan zich opnieuw aanmelden. De klant belandt dus terug bij stap 3.
- 4) De klant kan zijn inschrijving bevestigen.

9.9.1.3.2 Uitbreidingen

9.9.1.3.2.1 Aan agenda toevoegen

- 1) De klant ziet een knop waarmee hij de inschrijving aan zijn agenda kan toevoegen.
- 2) Na het indrukken van de knop wordt de agenda-applicatie op de smartphone geopend met een scherm voor een nieuwe afspraak. Alle gegevens over de opleiding zijn reeds ingevuld.

9.9.1.4 Een geïnteresseerde vindt een opleiding via de categorieën

Primaire actor: geïnteresseerde

Secundaire actoren: geen

Preconditie: De geïnteresseerde heeft de RealDolmen Education app geïnstalleerd op zijn smartphone.

Postconditie: De geïnteresseerde heeft de volgende informatie over de opleiding:

- Duur (aantal dagen)
- Beschrijving
- Inhoud van de opleiding
- Doelgroep
- Doel van de opleiding
- Vereiste voorkennis
- Methode waarop de opleiding gegeven wordt

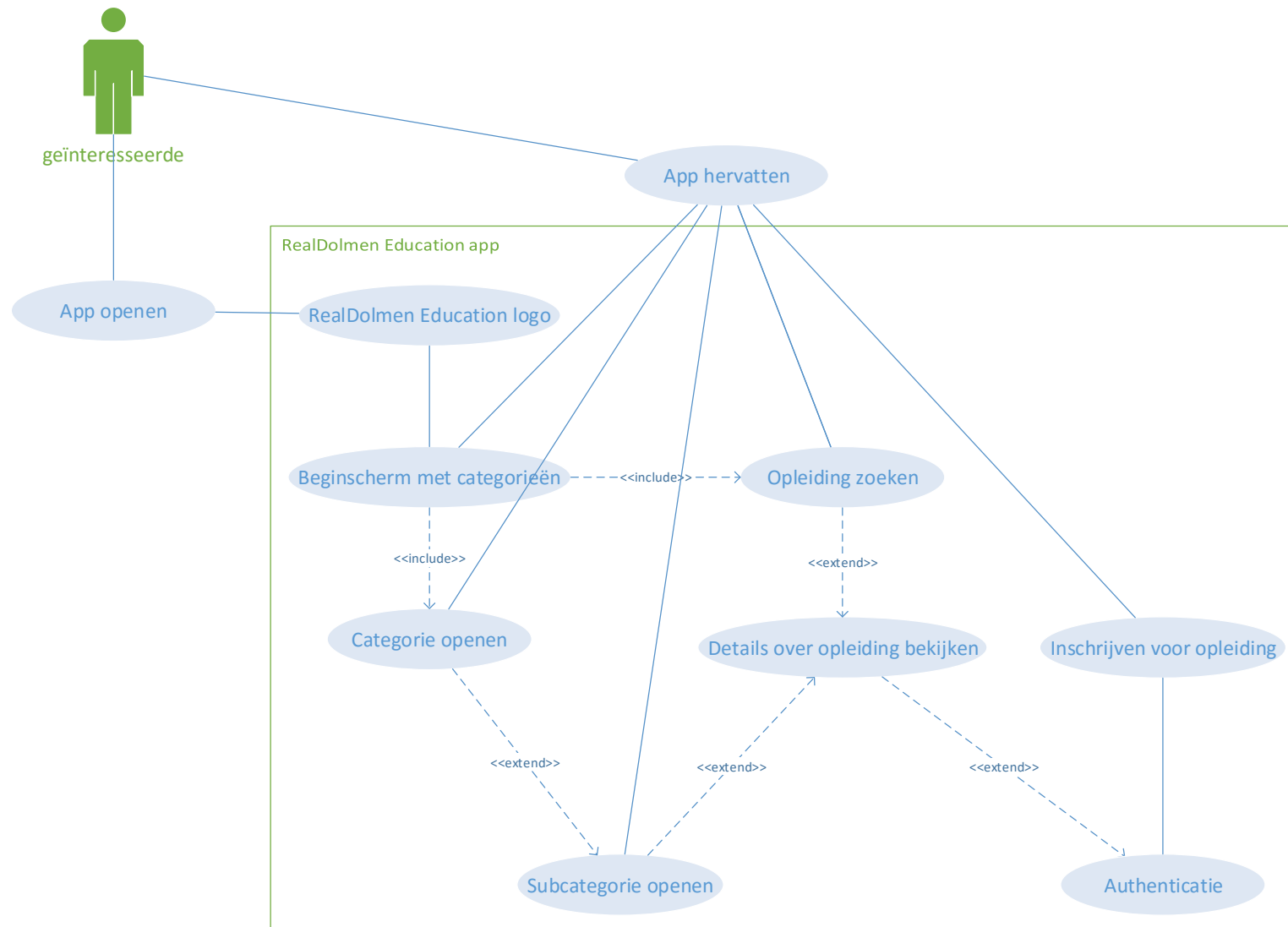
9.9.1.4.1 Hoofdscenario

- 1) De geïnteresseerde opent de RealDolmen Education app op zijn smartphone.
- 2) De geïnteresseerde wacht totdat de RealDolmen Education app geladen is, maar ziet ondertussen al het logo van RealDolmen (Education).
- 3) De geïnteresseerde ziet de lijst alle categorieën. Hier kan hij/zij door scrollen met zijn vinger.
- 4) De geïnteresseerde ziet een categorie die hem/haar interesseert en opent deze door erop te tikken.
- 5) De geïnteresseerde ziet alle subcategorieën die binnen de geselecteerde categorie ingedeeld zijn.
 - a. *De geïnteresseerde gaat terug met de Terugknop en keert terug naar stap 3.*
 - b. De geïnteresseerde tikt op een subcategorie die hem/haar interessant lijkt.
- 6) De geïnteresseerde ziet alle opleidingen die in de subcategorie zitten en kan ook hier scrollen.
 - a. *De geïnteresseerde gaat terug met de Terugknop en keert terug naar stap 5.*
 - b. De geïnteresseerde tikt op een opleiding die hem/haar interessant lijkt.
- 7) De geïnteresseerde krijgt de details over zijn opleiding te zien.
- 8) *Mogelijks moet er nog gescrold worden door met de vinger over het scherm te swipen.*

9.9.1.4.2 Uitbreidingen

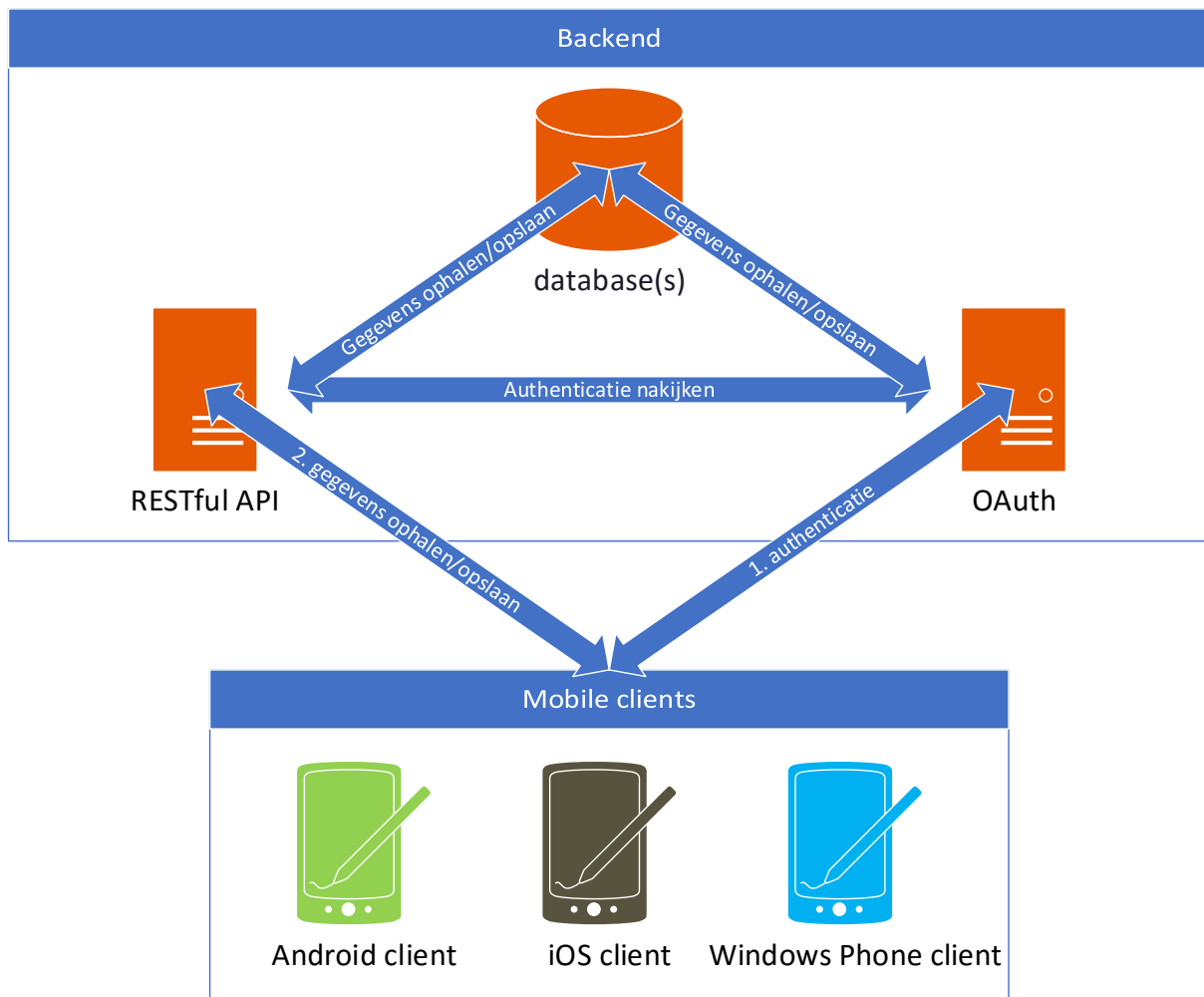
Dit scenario kan verder lopen in use case Een klant schrijft zich in voor een opleiding.

9.9.2 Diagram



9.10 Technische analyse

9.10.1 Architectuur



Figuur 6: algemeen overzicht van werking van mobile clients

De communicatie tussen de mobiele app en de backend verloopt bij elk mobiel platform op dezelfde manier. Er is een API voorzien die kan aangesproken worden om alle nodige gegevens op te halen of op te slaan indien van toepassing. Deze API heet de *EducationAPI*.

De authenticatie is opgesplitst, die zit niet bij in de API. Er is een aparte webserver voorzien die de authenticatie regelt. De authenticatie is vereist voor afgeschermden delen van de API. Voor deze afgeschermden delen heeft de eindgebruiker een gebruikersaccount nodig waarmee de gebruiker moet aanmelden voor hij/zij deze afgeschermden functies mag gebruiken. De gegevens-API en de authenticatie-API communiceren onderling om na te gaan of de authenticatie van de eindgebruiker in orde is. Er wordt dieper ingegaan op de authenticatie bij 9.10.3.4 OAUTH 2.

Voor de meeste endpoints⁹ is geen authenticatie nodig. Hierbij moet enkel stap 2 uit Figuur 6 uitgevoerd worden. Voor endpoints waarbij authenticatie vereist is, moet de mobiele app eerst de authenticatieprocedure uitvoeren met de authenticatie-API voordat de gegevens-API aangesproken kan worden.

⁹ Een endpoint op een API is een aanspreekpunt waarmee één specifieke functie kan uitgevoerd worden. Er kan bvb. een endpoint zijn voor het ophalen van een lijst van categorieën, een ander endpoint om de subcategorieën van een categorie op te halen en dergelijke.

9.10.2 Afhankelijkheden

9.10.2.1 *Xamarin*

Voor de Android app is Xamarin.Android nodig en voor iOS app is Xamarin.iOS nodig.

<https://xamarin.com/platform>

9.10.2.2 *Modellen*

Om eenvoudiger met de API te kunnen werken, worden dezelfde modellen gebruikt als bij de API (zie 9.10.3.2 API-modellen).

9.10.2.3 *Mapping*

Om eenvoudig objecten van de ene klasse om te zetten naar de andere klasse, wordt AutoMapper gebruikt.

<https://www.nuget.org/packages/AutoMapper/>

9.10.2.4 *Logging*

Om crashes en andere problemen met de app te kunnen analyseren, wordt Xamarin.Insights gebruikt. Hiermee worden statistieken en foutmeldingen teruggestuurd naar de ontwikkelaar van de app.

<https://www.nuget.org/packages/Xamarin.Insights/>

9.10.2.5 *Deserialization*

De antwoorden van de API zijn geformatteerd in JSON. Om deze makkelijker te kunnen uitlezen, kunnen de ASP.NET Client Libraries gebruikt worden.

<https://www.nuget.org/packages/Microsoft.AspNet.WebApi.Client/>

9.10.2.6 *Caching*

De antwoorden van de API kunnen tijdelijk opgeslagen worden om te vermijden dat deze meermaals opgehaald moeten worden. Dit principe heet caching en om de werking te vereenvoudigen, kan de client-module van CacheCow gebruikt worden. De API maakt zelf gebruik van de server-module van CacheCow.

<https://www.nuget.org/packages/CacheCow.Client/>

9.10.2.7 *MVVM*

Om de code een duidelijke structuur te geven, kan er gebruikt gemaakt worden van het Inversion of Control-principe (IoC) en een opdeling tussen applicatielogica en designlogica. Om dit te realiseren kan MVVM (Model-View-ViewModel) gebruikt worden.

<https://www.nuget.org/packages/MvvmLight/>

<https://www.nuget.org/packages/MvvmCross/>

9.10.2.8 *OAuth 2 helper*

Er zijn enkele nuget packages beschikbaar die authenticatie afzonderen van de rest van de code. Aangezien we OAuth 2 gebruiken en de authenticatieserver gebruikt maakt van ThinkTecture's IdentityServer (zie 9.10.3.4), is de gerelateerde client-package hier aangewezen.

<https://www.nuget.org/packages/Thinktecture.IdentityServer3.AccessTokenValidation/>

Bovenstaand package verzorgt enkel de authenticatie zelf, maar mist nog enkele functies die nodig zijn om vlot om te gaan met authenticatiecodes. Hiervoor kan IdentityModel gebruikt worden.

<https://www.nuget.org/packages/Thinktecture.IdentityModel>

9.10.2.9 Unit of Work- & Service-patterns

Alle communicatie van of naar de API kan afgescheiden worden van de rest van het systeem met designpatronen. Bij het Service-pattern is één klasse verantwoordelijk voor de communicatie. De achterliggende werking wordt hiermee volledig afgeschermd.

Om daarnaast deze Service-klasse overzichtelijk te houden, kan het Unit-of-Work-pattern gebruikt worden. Hierbij is elke methode uit de Service verantwoordelijk voor één taak. Die taak, de 'unit of work', kan individueel aangeroepen en uitgevoerd worden.

9.10.3 EducationAPI

9.10.3.1 API-endpoints

De documentatie van alle endpoints van de API is beschikbaar in een apart document [API_DOCS.HTML](#).

9.10.3.2 API-modellen



Figuur 7: overzicht van API-modellen

Op bovenstaande figuur is zichtbaar welke modellen in de API gebruikt worden. Ook de onderlinge relaties tussen de modellen zijn weergegeven.

9.10.3.3 REST

De API is een ASP.NET Web API. Alle endpoints zijn RESTful. REST staat voor Representational State Transfer. Het is geen gestandaardiseerd protocol op zich, maar het maakt gebruik van bekende standaarden. Hierbij worden meerdere beperkingen opgelegd aan de werking en het softwarematige design van de API. Het doel is om de werking van de API eenduidig te maken.

REST is een concept van Roy Fielding. Voor meer informatie hierover kan men op zijn website terecht: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

9.10.3.4 OAuth 2

Voor enkele functies van de applicatie moet een gebruiker zich kunnen identificeren. De persoonlijke gegevens van de gebruiker zijn nodig om een gebruiker te kunnen inschrijven voor een cursus of om hem/haar feedback te kunnen laten geven over een les. Deze persoonlijke gegevens kunnen opgehaald worden als de API weet wie de gebruiker is, met andere woorden als de gebruiker geïdentificeerd en dus aangemeld is met een persoonlijk gebruikersaccount.

9.10.3.4.1 Authenticatie

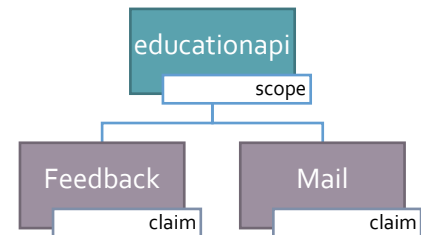
De eerste stap is de gebruiker identificeren. Deze stap heet authenticatie en omvat het aan- en afmelden van een gebruiker. Hierbij wordt een fysieke gebruiker dus gekoppeld aan een gebruikersaccount.

Het aan- en afmelden gebeurt op een gestandaardiseerde manier die OAuth 2 heet. OAuth 2 is een vrij flexibele standaard. Ontwikkelaars zijn relatief vrij in de implementatie ervan. De implementatie die de EducationAPI gebruikt is die van ThinkTecture. Er is een aparte authenticatieserver opgezet met behulp van IdentityServer (het product van ThinkTecture).

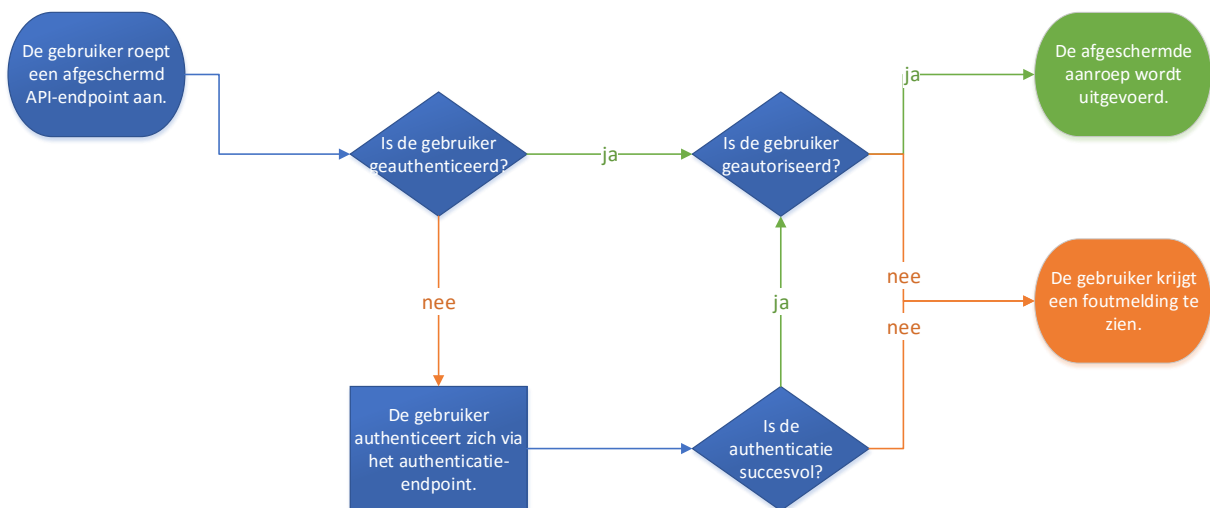
9.10.3.4.2 Autorisatie

De tweede stap is de gebruiker toegang verlenen tot de afgeschermd API-endpoints. Deze stap heet autorisatie. Hierbij wordt nagekeken welke rechten de aangemelde gebruiker heeft. Er wordt dus bij de oproepen naar afgeschermd API-endpoints nagekeken of de aangemelde gebruiker geautoriseerd is om dit endpoint te gebruiken.

OAuth 2 heeft hiervoor zogenaamde scopes. Een scope beschrijft een verzameling rechten die verleend kunnen worden aan gebruikers. Om gebruik te maken van de EducationAPI is de scope *educationapi* nodig. Deze scope is nog eens onderverdeeld in claims. Een claim omschrijft wat een gebruiker binnen een scope kan doen. De claims die bestaan binnen de *educationapi* scope zijn *Mail* en *Feedback*. De claims worden respectievelijk gebruikt om zich in te schrijven voor een cursus en om feedback te geven over een les.



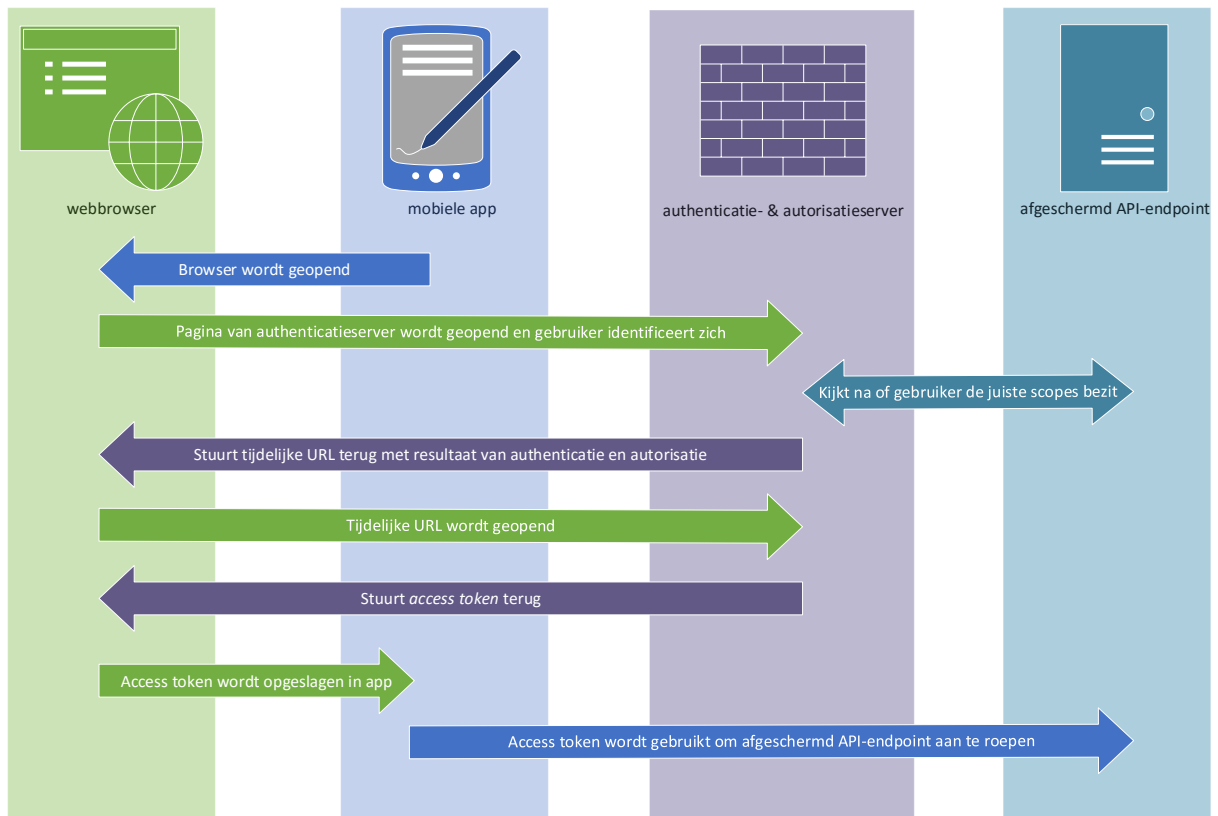
Figuur 8: indeling van scopes en claims



Figuur 9: authenticatie en autorisatie

9.10.3.4.3 In de praktijk

Om dit hele proces te bewerkstelligen maakt OAuth 2 gebruik van tokens. Deze tokens zijn codes met beperkte geldigheid die telkens voor een stap uit het proces in Figuur 9 gebruikt worden. Figuur 10 verduidelijkt hoe de uitwisseling van tokens verloopt.



Figuur 10: verloop van authenticatie en autorisatie

9.11 Vergelijking cross-platform development environments

9.11.1 Features per ontwikkelplatform

eigenschap	native op elk platform	Xamarin klassiek	Apache Cordova	Xamarin.Forms	Appcelerator Titanium	Adobe PhoneGap
Android	Ja	Ja	Ja	Ja	Ja	Ja
iOS	Ja	Ja	Ja	Ja	Ja	Ja
Windows Mobile	Ja	Ja	Ja	Ja	Ja	Ja
Windows desktop	Ja	Ja	Ja	Nee	Nee	Nee
native code	Ja	Ja	Nee	Ja	Nee	Nee
native designs	Ja	Ja	Nee	Nee	Nee	Nee
OS X	Ja	Ja	Nee	Nee	Nee	Nee
BlackBerry OS	Ja	Nee	Ja	Nee	Nee	Nee
Firefox OS	Ja	Nee	Ja	Nee	Nee	Nee
Ubuntu	Ja	Nee	Ja	Nee	Nee	Nee
web	Ja	Nee	Nee	Nee	Ja	Nee
>50% shared code	Nee	Ja	Ja	Ja	Ja	Ja
gecombineerd testen	Nee	Ja	Nee	Ja	Ja	Nee
eenvoudige IDE	Nee	Ja	Nee	Ja	Nee	Nee
totaal	11	10	8	7	6	4

9.11.2 Core features per ontwikkelplatform

eigenschap	Xamarin.Forms	Xamarin klassiek	Appcelerator Titanium	native op elk platform	Adobe PhoneGap	Apache Cordova
verhouding shared code	3	1	2	0	2	1
Android	Ja	Ja	Ja	Ja	Ja	Ja
iOS	Ja	Ja	Ja	Ja	Ja	Ja
Windows Mobile	Ja	Ja	Ja	Ja	Ja	Ja
gecombineerd testen	Ja	Ja	Ja	Nee	Nee	Nee
native code	Ja	Ja	Nee	Ja	Nee	Nee
eenvoudige IDE	Ja	Ja	Nee	Nee	Nee	Nee
native designs	Nee	Ja	Nee	Ja	Nee	Nee
totaal	9	8	6	5	5	4

9.12 Vergelijking native en web apps

9.12.1 Inleiding

Dit document verduidelijkt het verschil tussen een native app en een web of hybride app op smartphones.

Dit verschil is vooral een technisch verschil dat voor een eindgebruiker niet meteen visueel duidelijk is. Toch zijn native en web apps structureel verschillend op een manier die impact op de gebruikerservaring kan hebben.

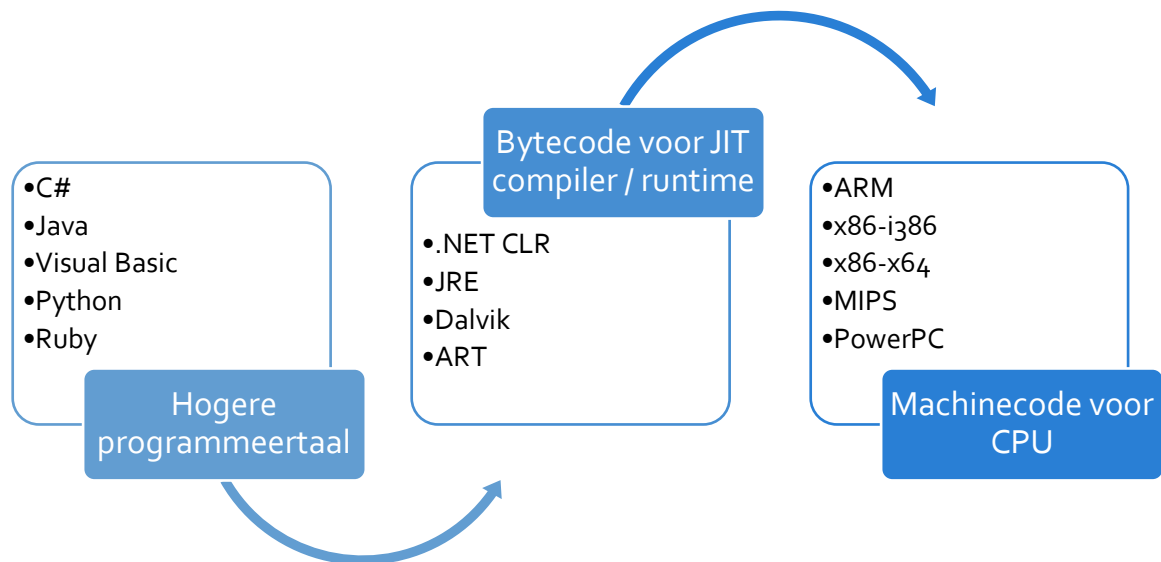
9.12.2 Wat is een native app?

9.12.2.1 Processors en programmacode

Een smartphone bevat net als een computer een processor, oftewel CPU (Central Processing Unit). Deze processor voert alle instructies in programmacode uit die ervoor zorgen dat het systeem werkt. Elke actie die het computersysteem moet uitvoeren wordt vertaald in programmacode die de processor verwerkt waarna de programmacode met het resultaat nieuwe berekeningen opbouwt.

Er zijn verschillende soorten programmacode. Machinecode is code die een processor onmiddellijk kan verwerken. Hoe de machinecode eruit moet zien, hangt af van het type processor. Zo werken hedendaagse desktopcomputers met een x86-processor en smartphones met een ARM-processor. Van beide processors bestaan ook varianten. De machinecode van elk type processor kan enkel gelezen worden op dat specifiek type processor, zo kan bijvoorbeeld machinecode voor een x86-processor enkel uitgevoerd worden op een x86-processor.

Heel wat computersystemen bevatten ook een JIT (Just in Time) compiler (heet soms ook runtime) die bytecode kan verwerken. Bytecode is een tweede soort programmacode die op een dergelijk abstracte manier geschreven is dat deze – in tegenstelling tot machinecode – op verschillende types processors gebruikt kan worden. Een JIT compiler leest deze bytecode tijdens de uitvoering en zet deze om in machinecode die een processor vervolgens uitvoert. Deze 'live' compilatie is dus een extra stap bij de uitvoering van programmacode. Het is ook mogelijk om de bytecode vooraf te compileren – om te zetten – naar machinecode. Bekende runtimes of JIT compilers zijn de Java Virtual Machine (Java Runtime Environment) van Oracle, de .NET Compiler Runtime van Microsoft, de Dalvik Virtual Machine van Google en de Android RunTime (ART) van Google.



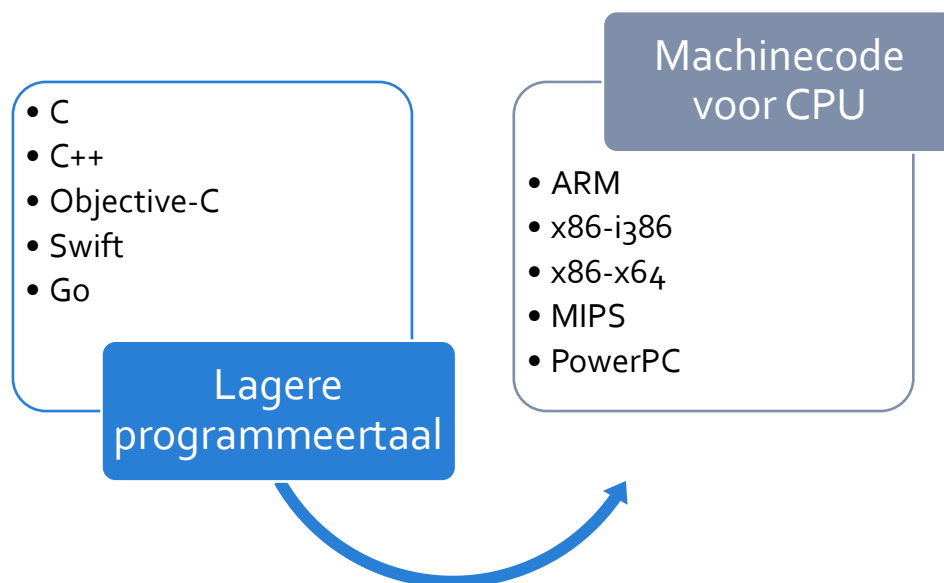
Figuur 11: Proces van compileren van programmacode met voorbeelden

9.12.2.2 Softwareontwikkeling

Bij de ontwikkeling van software wordt vaak een programmeertaal gebruikt die een compiler nog moet omzetten naar bytecode of machinecode. De gebruikte programmeertaal is begrijpelijker voor een softwareontwikkelaar en een compiler is verantwoordelijk om de gegenereerde byte- of machinecode zo efficiënt mogelijk te maken.

Andere programmeertalen zijn in staat om meteen gecompileerd te worden naar machinecode. Programmeertalen die naar bytecode compileren worden door Microsoft en andere grote softwarebedrijven vaak managed code genoemd en programmeertalen die naar machinecode compileren worden vaak unmanaged code of native code genoemd. In deze laatste variant zijn voor de softwareontwikkelaar vaak meer mogelijkheden beschikbaar, maar is de ontwikkeling van dergelijke code veel moeilijker.

In dezelfde analogie worden programmeertalen die eerst nog gecompileerd moeten worden hogere programmeertalen en programmeertalen die veel dichter bij de CPU werken lagere programmeertalen genoemd.



Figuur 12: Proces van uitvoering van native code met voorbeelden

9.12.2.3 Relatie tussen programmacode en native apps

Native apps zijn applicaties waarvan alle programmacode (na compileren) beschikbaar is in bytecode of machinecode. Die eerste variant waarbij bytecode gegenereerd wordt heet managed apps. De apps die naar machinecode gecompileerd worden, heten unmanaged apps. De term 'native apps' staat in principe voor beide categorieën, maar het komt voor dat een bedrijf met native apps enkel de laatste categorie van unmanaged apps bedoelt.

9.12.3 Wat is een web of hybride app?

9.12.3.1 De revolutie van HTML 5

Na Steve Jobs' open brief over het verouderde Adobe Flash platform van april 2010 begonnen de grotere internationale internetbedrijven zich te focussen op de ontwikkeling van HTML 5, de nieuwe versie van de programmacode van het web. Het web moest mobieler worden.

Het resultaat hiervan is HTML 5. HTML is de opmaaktaal waarin websites geschreven worden, maar de specificatie van HTML 5 omvat ook vernieuwingen op vlak van JavaScript, een programmeertaal die gebruikt wordt om websites interactief te maken. Het laat webontwikkelaars toe om gebruik te maken van meer mogelijkheden en sensoren van mobiele toestellen zoals locatiebepaling, accelerometrie¹⁰, camera's, offline opslag en andere sensoren.

9.12.3.2 Mobile web apps

Waarom zijn mobiele applicaties dan nog nodig? Op het moment van de publicatie van HTML 5 in 2014 waren smartphone-eigenaars het al gewoon dat websites niet optimaal werken op smartphones en dat enkel apps een optimale ervaring kunnen bieden. De mobiele platformen (Android, iOS, Windows Phone...) werden ook steeds uitgebreider en konden gebruikt worden om heel knappe mobiele applicaties te ondersteunen die met HTML 5 gewoonweg nog niet mogelijk waren.

Het grote nadeel van mobiele applicaties is dat ze op één mobiel platform werken. Voor elk mobiel platform moest een nieuwe applicatie ontwikkeld worden. Een web app (website geoptimaliseerd voor mobiel ge-

¹⁰ Meten van beweging. In welke richting beweegt het toestel? Hoe is het gedraaid (vlak/rechtop)? Ligt het neer? Wordt het bewogen? Aan welke snelheid? Vaak wordt deze functie gecombineerd met het kompas.

bruik) werkte steeds op alle mobiele platformen. Het enige wat de eindgebruiker nodig had was een webbrowser met ondersteuning voor HTML 5. Veel bedrijven zagen er dus toch het nut van in om een mobile web app te ontwikkelen.

Een mobile web app heeft zoals eerder vermeld ook nadelen ten opzichte van echte (native) mobiele applicaties. De technologie staat minder ver, de ontwikkeling ervan verloopt niet zo eenvoudig, niet alle webbrowsers ondersteunen HTML 5 volledig en er doen zich vaak problemen voor met offline opslag. Daarnaast moet een eindgebruiker ook nog steeds een webbrowser openen en naar een website surfen in plaats van op een icoontje op zijn of haar smartphone te drukken.

Vooraf om die laatste reden werd de hybride app ontworpen. Bij een hybride app wordt de mobiele website als het ware ingepakt. Vervolgens gaat men een heel eenvoudige native app bouwen die enkel en alleen bestaat uit een WebView. Een WebView is een component die ontwikkelaars kunnen gebruiken om in hun app een webpagina weer te geven. Zo lijkt de gebruiker een native app te hebben, maar wordt er uiteindelijk nog altijd een web app weergegeven.

9.12.4 Vergelijking

9.12.4.1 Welke platformen?

Hier wordt getracht een bondige vergelijkende analyse weer te geven van de kenmerkende verschillen tussen native apps en hybride apps. Veel eigenschappen van hybride apps doen zich namelijk ook voor bij web apps, maar bij die laatste is het visuele verschil voor de eindgebruiker minder groot.

9.12.4.2 Verschil in performantie en technische capaciteiten

Zoals eerder vermeld kunnen native apps hun programmacode vrij direct door een processor laten uitvoeren, zonder verdere compilatie of interpretatie. Web apps zijn daarentegen geschreven in JavaScript en HTML. Een JavaScript engine moet deze code nog interpreteren en omzetten naar byte- of machinecode.

De performantie van JavaScript hangt dus van deze engine af en of deze al dan niet rechtstreeks instructies naar de CPU kan sturen. Dit hele proces verloopt langzamer dan de uitvoering van native apps. Door de vertaalprocessen is de kans op fouten ook meteen groter.

Dit hele proces verloopt langzamer dan de uitvoering van native apps. Door de vertaalprocessen is de kans op fouten ook meteen groter.

Hoe dichter code bij de processor ligt, hoe meer mogelijkheden van de processor aangesproken kunnen worden. Daarnaast kunnen de instructies in de programmacode efficiënter opgebouwd worden. Beide factoren zorgen ervoor dat ontwikkelaars minder efficiënte en uitgebreide functies kunnen aanspreken in web apps.

9.12.4.3 Verschil in user experience

De gebruikservaring of user experience geeft aan hoe een eindgebruiker het gebruik van een app in zijn geheel ervaart. Dit is sterk gerelateerd aan het mobiele platform waar de eindgebruiker voor gekozen heeft. Elk mobiel platform heeft namelijk unieke kenmerken. Zo heeft een toestel met iOS altijd een fysieke knop die de gebruiker naar het beginscherm brengt, maar geen knop die de actie 'Terug' symboliseert. Die laatste is er wel op Windows Phone en Android, maar enkel Windows Phone heeft altijd een fysieke knop die actie 'Zoeken' weergeeft. De visuele weergave op elk platform verschilt ook sterk. Windows Phone is tweedimensionaal met weinig, maar felle kleuren.

Om de user experience te bevorderen, is het dus aangeraden om dezelfde trant te volgen van het mobiele platform waarop de applicatie moet draaien. Aangezien een web app op meerdere platformen werkt, is het onmogelijk om alle unieke kenmerken van de verschillende mobiele platformen te belichamen.

9.12.4.4 *Economisch verschil*

De ontwikkeling van een mobiele applicatie is duur. Aangezien er bij native apps (bijna altijd) een nieuwe app ontwikkeld moet worden voor elk mobiel platform, verhoogt dit aanzienlijk de ontwikkeltijd en de kosten. Een web app lost dit probleem op aangezien enkel het native gedeelte (de zogenaamde WebView) opnieuw ontwikkeld moet worden.

9.12.5 Gulden middenweg

Heel wat bedrijven zoals Adobe en Apache hebben producten gelanceerd die de ontwikkeling van apps voor meerdere platformen vereenvoudigt. Hun software zorgt ervoor dat de softwareontwikkelaar enkel een web app moet maken die daarna volledig automatisch omgezet wordt naar apps voor de verschillende mobiele platformen. Qua intentie is dit uitstekend, maar vaak is het resultaat een web app die op geen enkel mobiel platform degelijk werkt.

In 2011 kwam er een doorbraak bij de oprichting van het bedrijf Xamarin. Dit bedrijf is erin geslaagd om native apps te ontwikkelen die op meerdere platformen werken. Vandaag bieden ze een ontwikkelplatform aan waarbij apps kunnen ontwikkeld worden voor Android, iOS, OS X en Windows (Phone).

De softwareontwikkelaar schrijft code in C#. Vervolgens zet Xamarin deze code om naar native apps voor elk platform. Het is daarnaast ook mogelijk om het design (visueel) van de applicatie gelijk te trekken voor alle platformen, maar hier is de softwareontwikkelaar volledig vrij in.

9.12.6 Conclusie

Wanneer een bedrijf kiest voor de ontwikkeling van een mobiele applicatie, moeten verschillende belangrijke factoren in overweging genomen worden. Zo moet het budget afgewogen worden ten opzichte van de gevraagde gebruikerservaring.

Ook de mobiele platformen waarop een bedrijf actief wil zijn, kunnen doorslaggevende factoren zijn. Het spreekt voor zich dat een web app goedkoper is, maar kwalitatief moet onderdoen voor native apps. Deze stelling gaat logischerwijze enkel op in het ideale geval waarbij de softwareontwikkelaars de geschikte competenties hebben om ofwel web apps te ontwikkelen ofwel native apps te creëren die geoptimaliseerd zijn voor de mobiele platformen waar het bedrijf voor gekozen heeft.

Ten slotte kunnen ontwikkelomgevingen zoals die van Xamarin een gulden middenweg bieden. De toekomst moet uitwijzen of dit relatief jonge product aanslaat en een kwalitatieve oplossing is.

9.13 Vergelijking Xamarin & Xamarin.Forms

9.13.1 Introductie tot Xamarin

Xamarin is een verzameling producten van Xamarin Inc. dat softwareontwikkelaars een mogelijkheid biedt om native¹¹ apps te ontwikkelen die op meerdere mobiele platformen werken. De platformen die ze ondersteunen zijn de volgende:

- Google Android
 - Smartphones
 - Tablets
 - Smartwatches (Android Wear)
- Apple iOS
 - Apple iPhone
 - Apple iPad
 - Apple Watch
 - Apple TV
- Apple OS X
- Microsoft Windows
 - Windows 8 en Windows 8.1: desktop en tablet
 - Windows 10: desktop, tablet en smartphone
- Microsoft Windows Phone (8 of 8.1).

¹¹ Voor meer informatie over native apps, zie 9.12 VERGELIJKING NATIVE EN WEB APPS.



Figuur 13: opzet van Xamarin

Deze verzameling producten kunnen gezien worden als één volledig ontwikkelplatform. Er is integratie voorzien met Microsoft Visual Studio en de ondersteunde programmeertaal is C#. Daarnaast bieden ze zelf ook de ontwikkelomgeving Xamarin Studio aan en hebben ze hulpmiddelen voorzien zoals Xamarin.Insights. Hiermee kunnen softwareontwikkelaars statistieken ontvangen over problemen met de applicaties die ze geschreven hebben.

De opzet van Xamarin is om met zo weinig mogelijk code, zoveel mogelijk te doen. Het stelt softwareontwikkelaars in staat om code te hergebruiken voor hun apps op de verschillende mobiele platformen.

9.13.2 Ontwikkeling van mobiele applicaties

9.13.2.1 *Android*

Applicaties voor Android worden normaliter in de programmeertaal Java geschreven. Ontwikkelaars kunnen gebruikmaken van ingebouwde functies in Java of bijkomende API¹²'s in het Android-systeem.

De Java-code wordt omgezet naar bytecode¹³ en de Android RunTime (ART) zet deze om naar machinecode die een processor tot slot kan uitvoeren.

9.13.2.2 *iOS*

iOS-applicaties zijn geschreven in Swift of Objective-C. Beide programmeertalen kunnen door elkaar gebruikt worden. Ontwikkelaars kunnen ingebouwde functies uit Objective-C of Swift gebruiken of bijkomende API's in Cocoa Touch aanspreken. Dit laatste is een verzameling API's waarmee een softwareontwikkelaar componenten van het iOS-systeem kan aanspreken.

De Objective-C- of Swift-code wordt na de ontwikkeling meteen omgezet naar machinecode die uitgevoerd kan worden door de processor van een iPhone of een iPad.

Het design van iOS-apps gebeurt normaliter in het StoryBoard-formaat.

9.13.2.3 *Windows Store / Windows Phone*

Bij de lancering van Windows 8 introduceerde Microsoft een nieuwe vorm van applicaties voor desktops, laptops en tablets: Windows Store apps. Dit zijn moderne applicaties die geschreven kunnen worden in C# of Visual Basic waarbij het design in XAML gemaakt wordt. Als alternatief kunnen softwareontwikkelaars de talen vervangen door een combinatie van respectievelijk JavaScript en HTML met CSS.

De C#-code wordt door een compiler omgezet naar Common Intermediate Language (CIL), een vorm van bytecode. Wanneer de app opgestart wordt, wordt de CIL-code ingelezen door de Common Language Runtime (CLR). De CLR zet de CIL om naar machinecode die door een processor uitgevoerd kan worden.

Vanaf versie 8.1 hanteert Windows Phone hetzelfde principe.

Bij versie 10 heeft Microsoft gekozen voor Universal Windows Apps. Deze worden op dezelfde manier opgebouwd, maar werken meteen op alle soorten apparaten waarop Windows 10 draait: desktops, laptops, tablets, smartphones, Xbox, PixelSense¹⁴...

¹² API of Application Programming Interface: toegangspunt dat softwareontwikkelaars gebruiken om beroep te doen op bestaande functionaliteiten of gegevens.

¹³ Voor meer informatie over bytecode, zie 9.12 VERGELIJKING NATIVE EN WEB APPS.

¹⁴ Tafel met aanraakoppervlak

9.13.3 Onderscheid tussen de Xamarin-producten

9.13.3.1 *Xamarin.Android*

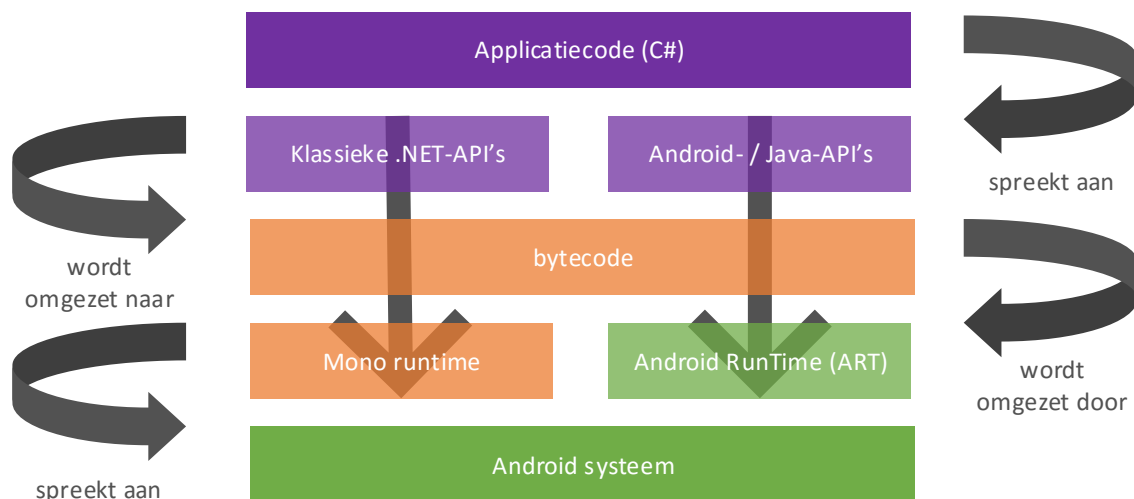
9.13.3.1.1 Code

Xamarin.Android bestaat uit een verzameling codebibliotheken die toegang geven tot de onderliggende codebibliotheken die gebruikt kunnen worden bij de ontwikkeling van Android apps.

Om de ontwikkeling te vereenvoudigen voor C#-programmeurs (.NET-programmeurs), zorgde Xamarin ervoor dat bestaande API's uit het .NET framework ook beschikbaar zijn in Xamarin.Android. Concreet komt dat erop neer dat een API zoals *System.Net.HttpClient* ook bestaat in Xamarin.Android. Programmeurs zijn het gewoon om deze *HttpClient* te gebruiken om communicatie met het internet te voorzien. Ze kunnen dan in Xamarin.Android de vertrouwde *HttpClient* op dezelfde manier gebruiken, maar Xamarin zorgt ervoor¹⁵ dat onderliggend de HTTP-functionaliteiten van Android aangesproken worden in plaats van die in het .NET framework.

Softwareontwikkelaars hebben dus de keuze. Ofwel gebruiken ze de API's die ze kennen vanuit het .NET framework, ofwel verwijzen ze via C# naar de in Android aangeboden API's.

Die keuze wordt verder in het systeem doorgetrokken. De uiteindelijke app wordt uitgevoerd door twee runtimes. Die runtimes zijn elk verantwoordelijk om de voor hen bestemde bytecode uit te voeren. Enerzijds is er de Mono Runtime die de bytecode uitvoert afkomstig van de klassieke .NET-API's. Anderzijds zorgt de Android RunTime voor de uitvoering van code die de Android- en Java-API's aanspreekt. De runtimes zelf zorgen dan voor verdere omzetting van code die het Android-systeem kan uitvoeren.



Figuur 14: van applicatiecode naar oproepen in het Android-systeem

9.13.3.1.2 Design

Er bestaan verschillende manieren om het design van de applicatie te maken, maar er wordt altijd gebruik gemaakt van een .axml-bestand (Android XML). AXML is een opmaaktaal die uitsluitend gebruikt wordt voor het ontwerp van Android applicaties. Verschillende ontwikkelomgevingen kunnen met AXML overweg: Eclipse, IntelliJ, Android Studio... Xamarin heeft deze ondersteuning uitgebreid naar Xamarin Studio en Visual Studio.

¹⁵ Microsoft, niet Xamarin, heeft de *HttpClient* voor Xamarin ontwikkeld. Microsoft koos ervoor om de *HttpClient* beschikbaar te maken als PCL (zie 9.13.3.4 PORTABLE CLASS LIBRARY (PCL)) zodat zowel Windows als Windows Phone van dezelfde *HttpClient* gebruik kunnen maken. Door de architectuur van Xamarin konden ze Xamarin.Android en Xamarin.iOS ook gebruiken als targets zodat hun compilers hiermee overweg kunnen.

9.13.3.1.3 Ontwikkelervaring

Voor de ontwikkeling van Android applicaties levert Xamarin een ervaring af die ontwikkelaars veel mogelijkheden biedt. Hierbij is Xamarin erin geslaagd de complexe onderliggende architectuur te verbergen om het voor ontwikkelaars eenvoudiger te maken. Ze kunnen gebruik maken van een vertrouwde ontwikkelomgeving zoals Visual Studio, maar er worden ook alternatieven aangeboden zoals Xamarin Studio of Android Studio voor het design.

9.13.3.2 Xamarin.iOS

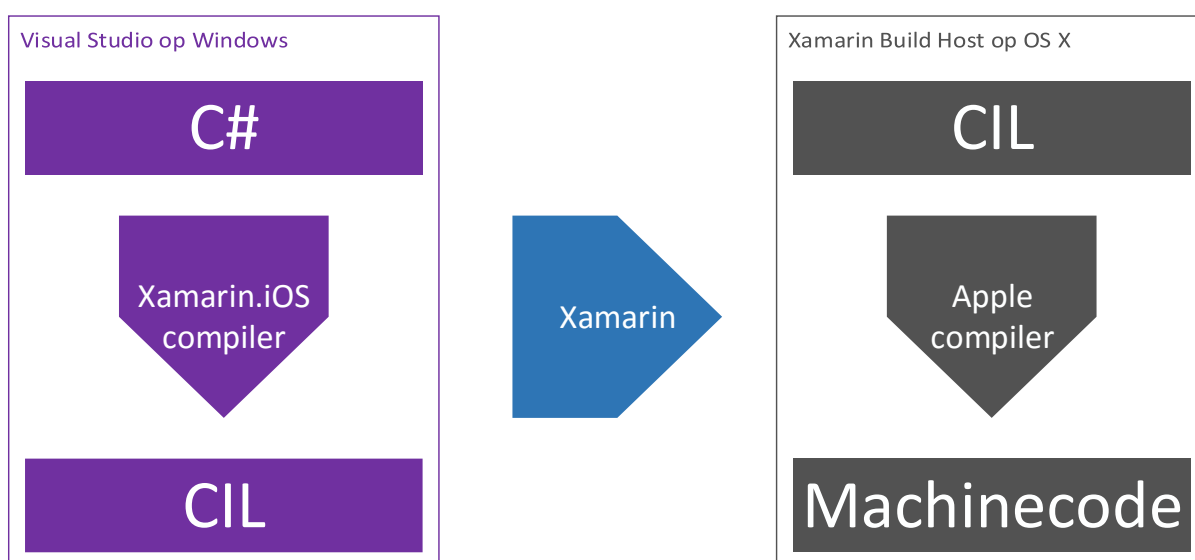
Xamarin.iOS levert – in sterk contrast met Xamarin.Android – een volledig andere ervaring voor softwareontwikkelaars af. Wegens door Apple ingebouwde beperkingen in iOS, verloopt de ontwikkeling met Xamarin heel anders.

9.13.3.2.1 Code

Het schrijven van de code verloopt wel gelijkaardig. Vertrouwde .NET-componenten zoals de eerder vermelde *HttpClient* werden gerecreëerd in Xamarin.iOS en kunnen door softwareontwikkelaars gebruikt worden op dezelfde manier als ze gewoon waren in het .NET framework. Daarnaast zijn er nog andere API's beschikbaar die een koppeling vormen naar bekende componenten in iOS.

De compilatie van de C#-code naar een bruikbare app verloopt volledig anders. Bij Xamarin.Android konden meerdere runtimes (Mono en ART) simultaan bytecode verwerken, maar bij iOS is dit niet mogelijk. iOS staat niet toe dat er een runtime actief is die code genereert. Daarom moet alle code van een iOS app op voorhand omgezet worden naar machinecode door een compiler.

De Xamarin.iOS compiler zet de C#-code om naar CIL¹⁶-code. Het gaat hier om een variant van de CIL die enkel in deze context gebruikt wordt. Deze CIL-code wordt vervolgens naar een toestel met OS X – het besturingssysteem van Apple voor Mac – gestuurd waarop de Xamarin Build Host geïnstalleerd staat. Deze Build Host roept de Apple compiler op die ingebouwd is in OS X. Die laatste gaat op zijn beurt de code omzetten naar machinecode voor iPhone, iPad...



Figuur 15: ontwikkelproces met Xamarin.iOS

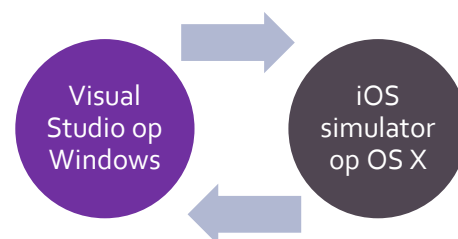
9.13.3.2.2 Design

Vanaf versie 4 kan Visual Studio aangewend worden voor het ontwerp van iOS apps. Daarvoor was dit enkel mogelijk in Xamarin Studio voor OS X of Xcode, de ontwikkelomgeving van Apple.

¹⁶ Zie 9.13.2.3 WINDOWS STORE / WINDOWS PHONE

9.13.3.2.3 Ontwikkelervaring

Alhoewel dit al verbeterd is sinds versie 4 (uitgebracht in november 2015), is de ontwikkelervaring met Xamarin.iOS vrij verwarrend en complex. De oorzaak hiervan is het complexe onderliggende proces. Doordat alle code uiteindelijk op een Mac terecht moet komen, is het onmogelijk om code lokaal te testen in Visual Studio. De code kan enkel uitgevoerd worden in een simulator op OS X. Ontwikkeling met Xamarin Studio voor Windows is zelfs volledig onmogelijk.



Figuur 16: ontwikkelervaring met Xamarin.iOS

9.13.3.3 Windows Phone / Windows Store

9.13.3.3.1 Code

Voor Windows heeft Xamarin geen bijkomende codebibliotheken die nodig zijn om apps te schrijven in C#. Apps voor Windows zijn namelijk sowieso al in C# geschreven¹⁷. Visual Studio en onderdelen ervan zoals MS Build zorgen voor de compilatie naar CIL die uitgevoerd kan worden op Windows (Phone).

9.13.3.3.2 Design

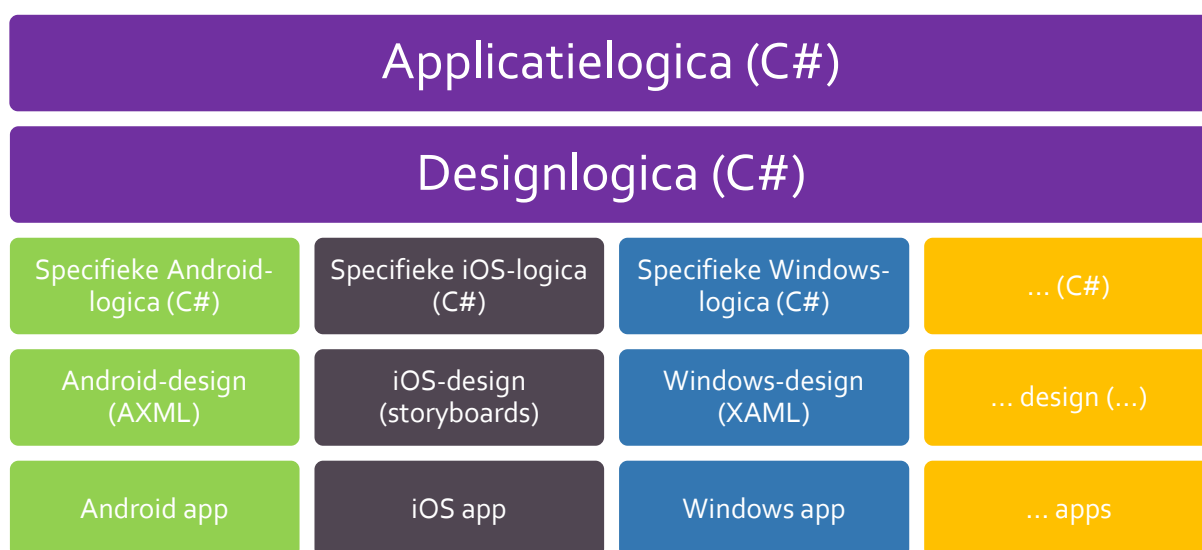
Zowel Windows Store als Windows Phone apps worden ontworpen in XAML.

9.13.3.4 Portable Class Library (PCL)

Er lijken dus heel wat verschillende compilers gebruikt te worden voor alle soorten projecten. Elk type project (Xamarin.Android, Xamarin.iOS, Windows Store apps...) gebruikt bijna een andere compiler. Toch is de bedoeling om zoveel mogelijk code te delen over de verschillende apps. Daartoe hebben Microsoft en Xamarin de Portable Class Library in het leven geroepen.

Een Portable Class Library is een codebibliotheek die door meerdere compilers (worden in deze context *targets* genoemd) gecompileerd kan worden. Een Xamarin.Android-project en een Xamarin.iOS-project kunnen dus allebei naar exact dezelfde PCL-code verwijzen. Bij de compilatie wordt de PCL dan een keer voor Android gecompileerd en een keer voor iOS.

Om dan zoveel mogelijk code te delen, wordt er getracht alle applicatielogica en de gemeenschappelijke designlogica in PCL's te plaatsen.



Figuur 17: structuur van een project met apps voor meerdere platformen en PCL's

¹⁷ Het is ook mogelijk om andere programmeertalen te gebruiken zoals Visual Basic, maar die zijn in deze context niet relevant.

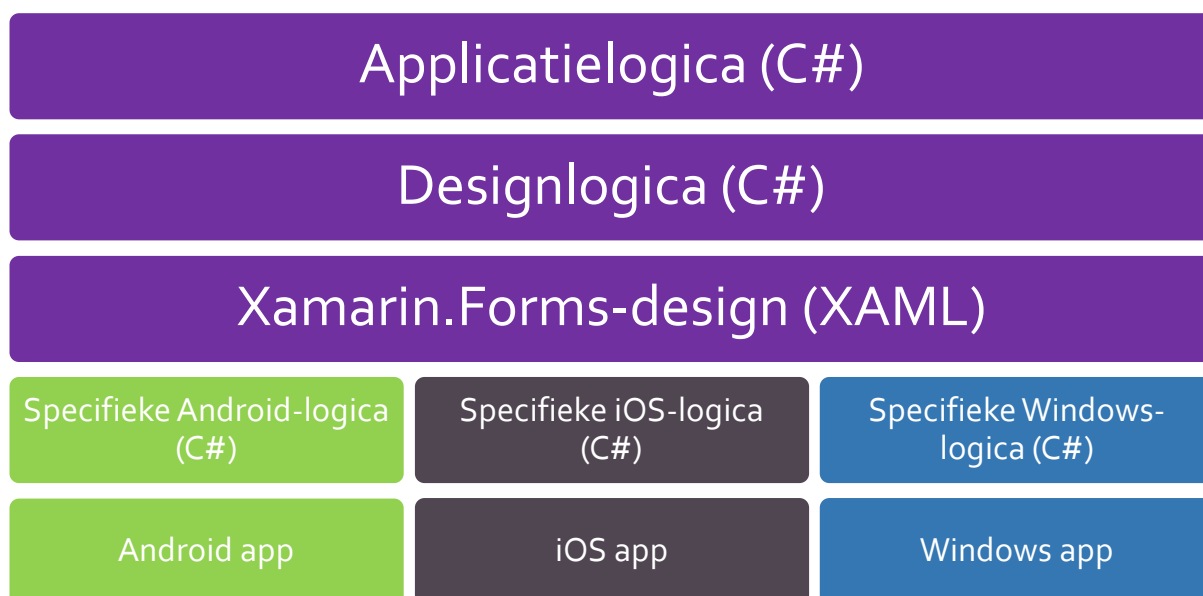
9.13.3.5 Xamarin.Forms

9.13.3.5.1 Structuur

In Figuur 17 is zichtbaar dat de twee bovenste lagen al volledig gedeeld worden over alle apps. Daaronder zitten twee lagen die specifiek zijn aan elke app en per mobiel platform opnieuw gemaakt moeten worden. Om ook op deze lagen code te delen, werd Xamarin.Forms in het leven geroepen.

Een Xamarin.Forms-project werkt analoog aan Xamarin.Android en Xamarin.iOS. Het bevat geen applicatielogica, de geschikte plaats hiervoor blijft een PCL. Het bevat daarentegen wel alle designlogica voor alle applicaties.

Het visuele design wordt ontworpen in XAML, dezelfde opmaaktaal die gebruikt wordt voor Windows Store / Phone apps. Het gaat hier om een variant van XAML die enkel bij Xamarin.Forms gebruikt kan worden. Analooq aan Windows Store apps, wordt er per scherm ook C#-code geschreven die de designlogica van dit scherm omvat.



Figuur 18: structuur van Xamarin.Forms

De 4^{de} laag in Figuur 18 met de specifieke logica per applicatie is meestal beperkt. Met Xamarin.Forms wordt code zoveel mogelijk gedeeld.

9.13.3.5.2 Code

Xamarin.Forms apps kunnen volledig ontwikkeld worden in Visual Studio en Xamarin Studio, met uitzondering van de iOS-component. Net zoals bij Xamarin.iOS is een Apple Mac vereist voor de compilatie van de iOS-app.

De API's die aangesproken kunnen worden, verschillen sterk van Xamarin.Android en Xamarin.iOS. Er werd namelijk getracht om één API te creëren die werkt over alle verschillende apps heen. Met die ene API kan een GPS-systeem aangesproken worden, accelerometrie en andere sensoren.... Xamarin zorgt er bij de compilatie voor dat onderliggend de juiste API voor elk platform opgeroepen wordt.

9.13.3.5.3 Design

Het design is voor elk mobiel platform hetzelfde. Met de aangepaste vorm van XAML kunnen schermen ontworpen worden die Xamarin bij de compilatie omzet naar aparte designs per mobiele app. Dit heeft zowel voor- als nadelen. 9.13.4.2 Design gaat hier verder op in.

9.13.4 Keuze maken tussen Xamarin (Xamarin.Android en Xamarin.iOS) en Xamarin.Forms

9.13.4.1 Code sharing

Op vlak van code sharing staat Xamarin.Forms verder dan de klassieke Xamarin (Xamarin.Android en Xamarin.iOS). Niet alleen wordt de applicatielogica gedeeld, maar ook alle designs en bijhorende logica worden gedeeld. Bij de meeste mobiele applicaties kost het design het meeste werk. Xamarin.Forms biedt dus zeker en vast een grote meerwaarde op vlak van code sharing.

9.13.4.2 Design

Het grote voordeel aan Xamarin.Forms is ook meteen een groot nadeel. Het gedeeld design kan dan wel eenvoudiger zijn in de ontwikkelfase, maar bij het gebruik ervan ondervinden gebruikers nadelen. Zoals beschreven in 9.6 REQUIREMENTS ANALYSE, is het belangrijk om de platformspecifieke user experience in het achterhoofd te houden. Zo hebben Android- en Windows-toestellen fysieke knoppen die de actie *terug* voorstellen, maar staat op iOS een soortgelijke knop links bovenaan op het scherm zelf. Er zijn talloze interacties die verschillen op elk platform. Omdat ze nagenoeg altijd consistent op dezelfde manier uitgewerkt zijn, wordt de uitvoering van dergelijke interacties een gewoonte voor de gebruiker. De gebruiker moet niet langer nadenken over hoe of waar hij of zij die interactie zou kunnen uitvoeren. De toepassing van deze platformspecifieke interacties komt de ervaring voor de gebruiker in zijn geheel ten goede. De applicatie wordt ervaren als eenvoudiger en voor de hand liggend.

Aangezien Xamarin.Forms eenzelfde design hanteert voor alle verschillende platformen – zonder differentiatie – wordt de platformspecifieke user experience genegeerd. Het is mogelijk om deze toch gedeeltelijk in te bouwen in een Xamarin.Forms-project, maar uiteindelijk kan de vraag gesteld worden of het dan nog wel efficiënter is om Xamarin.Forms te gebruiken in plaats van de klassieke Xamarin.

Ten slotte is een design met Xamarin.Forms niet even stabiel als een design specifiek voor een mobiel platform. Een gebruiker kan vertragingen opmerken of denken dat de app even niet reageert. Dit valt te wijten aan de extra stap bij de compilatie. Het design moet namelijk eerst nog vertaald worden voor elk mobiel platform.

9.13.4.3 Ontwikkelervaring

De ervaring die een softwareontwikkelaar krijgt, bepaalt mee de looptijd en de kost van een project. Als de ontwikkeling complex verloopt, verlengt dit de looptijd van het project of verlaagt dit de kwaliteit van het afgeleverde product.

De ervaring bij Xamarin.Forms is analoog aan die met de klassieke Xamarin-projecten. In beide gevallen verloopt de ontwikkeling van apps voor iOS complexer en lukken Android en Windows apps zonder problemen.

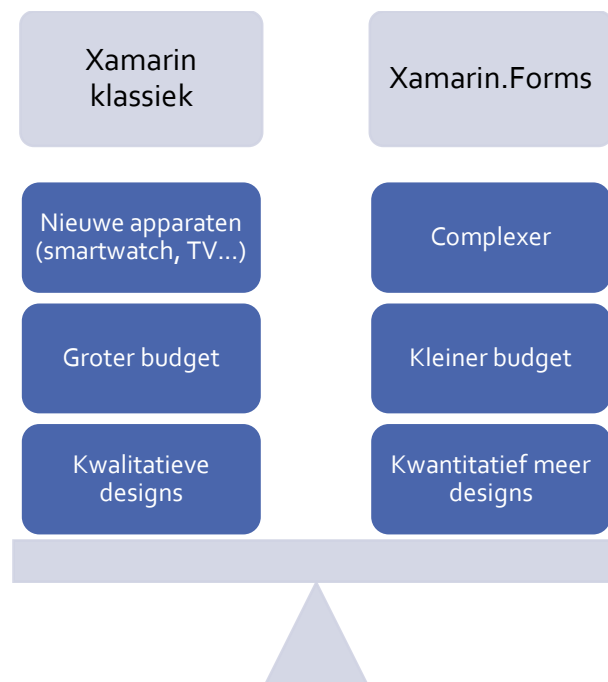
9.13.5 Conclusie

De keuze tussen Xamarin en Xamarin.Forms ligt niet voor de hand. Wanneer er voor een van beide gekozen werd, betekende dit op zich al dat de klant in principe een ervaring wilt die op elk platform degelijk werkt en budgettair genoeg ademruimte krijgt. Er zijn verschillende factoren die geanalyseerd moeten worden om een juiste keuze te kunnen maken.

Vooreerst is het belangrijk om de functionele en niet-functionele voorwaarden van de applicatie te onderzoeken. Bij een applicatie waar er veel designs nodig zijn, maar waarbij de kwaliteit van het design op zich minder belangrijk is, ligt de keuze voor Xamarin.Forms voor de hand. Vice versa zorgt de nood aan kwalitatieve designs voor de keuze voor klassieke Xamarin.

Als het budget groot is en de ervaring voor de gebruiker optimaal moet zijn, dan lijkt een keuze voor klassieke Xamarin logisch.

Tot slot is het mogelijk dat de keuze bepaald wordt door de mobiele platformen waarvoor applicaties ontwikkeld moeten worden. Het is namelijk volstrekt mogelijk om Xamarin te gebruiken om een applicatie voor één mobiel platform zoals Android of iOS te ontwikkelen. In dit geval is het overbodig om de bijkomende complexiteit van Xamarin.Forms te gaan gebruiken. Daarnaast ondersteunt Xamarin.Forms nog geen apps voor Apple TV, Apple Watch, Android Wear en andere modernere apparaten.



Figuur 19: criteria Xamarin / Xamarin.Forms

Zowel Xamarin klassiek als Xamarin.Forms evolueren snel. Daarom is een vergelijkende analyse hiervan niet meer dan een momentopname. Een jaar later kunnen de kaarten weer anders liggen waardoor de voorkeur voor een van beide varianten kan wijzigen.

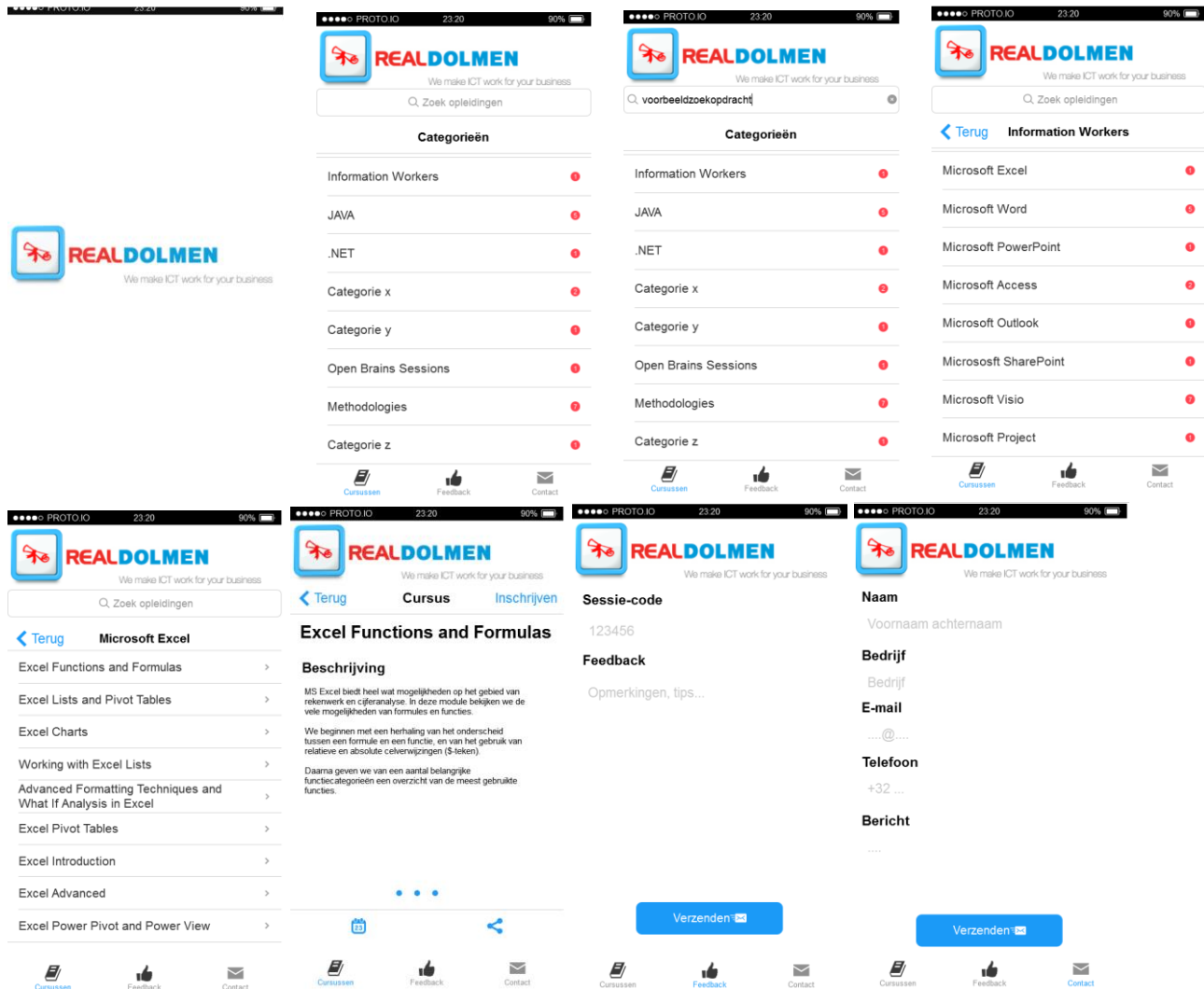
9.14 API documentatie

De documentatie van de API werd gegenereerd aan de hand van enkele tools die vaak gebruikt worden voor de documentatie van een API. In de sector wordt hiervoor meestal de tool Swagger gebruikt. Op de website van deze tool, swagger.io, staan verwijzingen naar integraties met andere bekende hulpmiddelen. Dit vereenvoudigt het documenteren aanzienlijk.

Swagger gaat zelf op zoek in de broncode van de API naar alle aanwezige functies. Softwareontwikkelaars kunnen tijdens het programmeren in de broncode zelf documentatie voorzien over de functies die ze inbouwen. Swagger gebruikt deze *in-code* documentatie en genereert vervolgens een gestructureerd bestand met alle functies en hun documentatie. Dit resultaat kan nadien weer ingelezen worden door andere tools die dan een leesbaar document kunnen opbouwen in verschillende formaten. Het meest courante formaat hierbij is een webpagina.

Deze webpagina is terug te vinden als bestand [API_DOCS.HTML](#).

9.15 Schermen van wireframes



De wireframes zelf kunnen geopend worden via het bestand [INDEX.HTML](#) met de browser Google Chrome.

9.16 Demo apps

Om te demonstreren hoe het mogelijk is om apps te maken voor meerdere mobiele platformen met dezelfde broncode, werden twee heel simpele applicaties ontwikkeld. Beide applicaties hadden dezelfde functies, maar de ene was in Xamarin (klassiek) en de andere in Xamarin.Forms ontwikkeld.

De applicaties maken gebruik van een API van de KU Leuven om gegevens over een opleiding op te halen. Zo kan de URL <https://dataservice.kuleuven.be/v1/program2015/search.json?q=programId:51854577> gebruikt worden om gegevens over de opleiding Toegepaste Informatica aan Odisee op te halen. Deze gegevens worden vervolgens verwerkt en weergegeven in de applicatie.

De Xamarin app werd ontwikkeld voor Android en voor Windows desktop. De Xamarin.Forms app is beschikbaar voor Android en Windows Phone.

De broncode van beide apps werd gepubliceerd op <https://github.com/SamuelDebruyn/Xamarin-Research>